Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки
(повне найменування інституту, факультету)

Кафедра автоматики та управління в технічних системах
(повна назва кафедри)

До захисту допущено
Завідувач кафедри
_____ Ролік О.І.__
(підпис) (ініціали, прізвище)
“___” _____2019_р.

Дипломний проект

на здобуття освітнього ступеня “бакалавр”
(назва ОС)

за напрямом 6.050201 “Системна інженерія”
(код та назва напряму підготовки)

на тему: Система моніторингу динаміки ринку

Виконав: студент __4__ курсу, групи ____ІА-52_____
(шифр групи)

Вакулка Тарас Сергійович _____ _____
(прізвище, ім’я, по батькові) (підпис)

Керівник _____аспірант Дорога-Іванюк О.О. _____ _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант _____ _____
(назва розділу) (посада, вчене звання, науковий ступінь, прізвище, ініціали) (підпис)

Рецензент _____ _____
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цьому дипломному проекті немає запозичень з праць інших авторів без відповідних посилань.

Студент _____
(підпис)

Київ – 2019_ року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки
(повна назва)

Кафедра автоматики та управління в технічних
(повна назва)

Освітній ступінь бакалавр
Напрям підготовки 6.050201 “Системна інженерія”
(шифр і назва)

ЗАТВЕРДЖУЮ
Завідувач кафедри
Ролік О.І. _____
(прізвище ініціали) (підпис)
“___”_____2019 р.

З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЕКТ СТУДЕНТУ

Вакулка Тарас Сергійович
(прізвище, ім’я, по батькові)

1. Тема проекту: Система моніторингу динаміки ринку,
керівник проекту: Дорога-Іванюк Олена Олександрівна,
(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)
затверджені наказом по університету від _06__ квітня __2019___ р. №___-с

2. Термін подання студентом проекту 15 червня 2019 р.
3. Вихідні дані до проекту
База даних прикладів проданих автомобілів.
4. Зміст проекту
Проведення загального аналізу ринку продажів автомобілів, моніторинг та прогнозування вартості автомобіля. Огляд ефективних методів побудови моделі прогнозування.
5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо)

6. Консультанти розділів проекту*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 5 березня 2019 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання дипломного проекту	Термін виконання етапів проекту	Примітка
1	Розробка, оформлення, узгодження та затвердження технічного завдання на дипломний проект	06.03.2019 – 16.03.2019	Виконано
2	Аналіз вимог завдання, вибір методів і засобів розв'язання поставленої задачі	17.03.2019 – 29.03.2019	Виконано
3	Аналіз структури та стратегії обробки даних	30.03.2019 – 15.04.2019	Виконано
4	Попередня обробка даних, виправлення відсутніх значень	16.04.2019 – 28.04.2019	Виконано
5	Дослідження даних, вибір алгоритму машинного навчання для побудови моделі прогнозування	29.04.2019 – 14.05.2019	Виконано
6	Побудова моделі прогнозування за допомогою алгоритмів лінійної та поліноміальної регресії. Оптимізація моделі	16.04.2019 – 31.05.2019	Виконано
7	Підготовка матеріалів до друку та оформлення пояснювальної записки	01.06.2019 – 09.06.2019	Виконано
8	Підготовка доповіді до захисту та оформлення ілюстративного матеріалу	10.06.2019 – 18.06.2019	Виконано

Студент _____ **Вакулка Т.С.**
(підпис) (прізвище та ініціали)

Керівник проекту _____ **Дорога-Іванюк О.О.**
(підпис) (прізвище та ініціали)

* Консультантом не може бути зазначено керівника дипломного проекту.

АНОТАЦІЯ

Вакулка Т.С. Система моніторингу динаміки ринку. КПІ ім. Ігоря Сікорського, Київ, 2019.

Проект містить: 76 с., 7 табл., 33 рис., 5 посилань на джерела.

Ключові слова: система моніторингу, аналіз даних, машинне навчання, алгоритм, регресія.

Об’єктом розробки є система моніторингу динаміки ринку.

Мета розробки - проведення аналізу ринку продажів автомобілів та побудова моделі для прогнозування цін на автомобілі, за для підвищення економічних показників ринку продажів автомобілів.

У дипломному проекті розроблено систему моніторингу динаміки автомобільного ринку. Останнім часом спостерігається бурхливий розвиток технологій, що дозволяють аналізувати великі об’єми даних. У ході виконання проекту було проведено огляд ефективних алгоритмів машинного навчання та методів обробки даних. За допомогою алгоритмів лінійної та поліноміальної регресії було побудовано модель прогнозування цін на автомобілі.

Отримані результати та побудована модель прогнозування дозволяють підвищити економічні показники компаній, які займається продажами автомобілів.

SUMMARY

Vakulka T.S. Market dynamics monitoring system. Igor Sikorsky KPI, Kyiv, 2019.

The project contains: 76 p., 7 tabl., 33 figures, 5 references to the source.

Keywords: monitoring system, data analysis, machine learning, algorithm, regression.

The object of development is a system for monitoring the dynamics of the market. The purpose of the development is to carry out an analysis of the car sales market and build a model for predicting car prices, for improving the economic performance of the car sales market.

The graduation project has developed a system for monitoring the dynamics of the automotive market. Recently, there is a rapid development of technologies that allow you to analyze large volumes of data. During the project implementation, an overview of effective algorithms of machine learning and data processing methods was conducted. With the help of linear and polynomial regression algorithms, a model for predicting car prices was constructed.

The obtained results and the built forecasting model allow to increase the economic indicators of companies engaged in car sales.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

LDA	Linear discriminant analysis - лінійний дискримінантний аналіз;
KNN	K-nearest neighbors - K-найближчих сусідів;
SVM	Support vector machine - метод опорних векторів;
ANSI C	American National Standards Institute – стандарт мови С;
IBM	International Business Machines Corporation;
SQL	Structured query language;
CSV	Comma-Separated Values;

ВСТУП

Сьогодні моніторинг ринку є невід'ємною частиною маркетингових досліджень. Застосовуються методи машинного опрацювання інформації і збираються дані за каналами офіційної статистики (на базі даних інформаційних обчислювальних систем), опрацьовується комерційна інформація, що з'являється в періодичних виданнях, рекламних проспектах та інших матеріалах; опрацьовуються дані, що надходять зі спеціалізованих підрозділів компаній, які збирають інформацію безпосередньо із ринку. На підставі одержуваних даних робляться висновки щодо основних тенденцій на ринку і прогноз перспектив. При цьому використовуються найрізноманітніші методи аналізу інформації: від суто математичних методів до експертних оцінок. Дослідження ринку є вагомим елементом маркетингового дослідження, що сприяє зменшенню невизначеності в частині прийняття комерційних рішень. Включає такі процедури: з'ясування розміру і характеру ринку, розрахунок реальної і потенційної місткостей ринку, аналіз факторів, що впливають на розвиток ринку (урахування специфічних особливостей аналізу товарного і регіонального ринку, установлення ступеня насичення ринку і т. п.), сегментація ринку і визначення типів споживачів за основними характеристиками: вік, стать, дохід, професія, соціальне становище, місце мешкання і т.д.

Актуальність проекту полягає у тому, що потреба у маркетингових дослідженнях для корпоративних мереж зростає, в переважній більшості для підвищення економічних показників або прийняття комерційних рішень, саме тому впровадження нових технологій моніторингу та аналізу даних займає велику роль, адже забезпечую можливість роботи з великим обсягом даних та має відносно високі показники точності.

Об'єктом досліджень є ринок автомобільних продажів.

Метою проекту є розробка системи моніторингу та аналізу динаміки автомобільного ринку. Загальний аналіз та порівняння методів машинного навчання для обробки даних проданих авто та прогнозування цін.

Для успішного виконання дипломного проекту необхідно мати базові знання з наступних дисциплін: програмування, машинне навчання, математична статистика, математичний аналіз.

1. ЗАГАЛЬНА МОДЕЛЬ СИСТЕМИ МОНІТОРИНГУ РИНКУ

1.1 Відомості про маркетинговий моніторинг

Для ефективного управління бізнесом і підвищення якості прийняття управлінських рішень необхідно володіти релевантною інформацією. На сучасному етапі розвитку економіки об'єктивно створилася потреба в нових підходах до її збору, обробки та аналізу. У світовій практиці такі підходи реалізуються за допомогою моніторингу.

Моніторинг - важливий етап в прийнятті рішень, так як він створює матеріал для аналітики і оперативного управління. Головна сфера практичного застосування моніторингу - інформаційне обслуговування управління в різних сферах діяльності.

Основною умовою функціонування моніторингу як інформаційної технології є глобальне охоплення аудиторії. Моніторинг різносторонній і всеосяжний, він втілює в собі цілу систему базових знань способів аналізу і обробки отриманої інформації, перш за все з метою створення прогнозів, необхідних для управління будь-якого рівня.

Аналіз різних поглядів на сутність і зміст маркетингового моніторингу показує, що склалися три принципово схожих методологічних підходи до його трактування: якісний, системний і прогностичний.

Практично всі автори сходяться в тому, що маркетинговий моніторинг пов'язаний з системою повторних спостережень в просторі і в часі, з певними цілями, як правило, відповідно до заздалегідь розробленою програмою. Моніторинг проводиться на основі систематичного збору і обробки інформації, додаткових досліджень, діагностики стану і тенденцій розвитку конкретного об'єкта. У більшій частині визначень виділяється результативна стадія

моніторингу - вироблення рекомендацій щодо прийняття управлінських рішень та вдосконалення політик, програм і заходів.

Систематизація поглядів вчених і практиків в області аналізу і управління ринковими процесами і моніторингом товарних ринків в системі маркетингового аналізу дозволяє нам дати інтегральний підхід до визначення маркетингового моніторингу (Рис. 1.1).

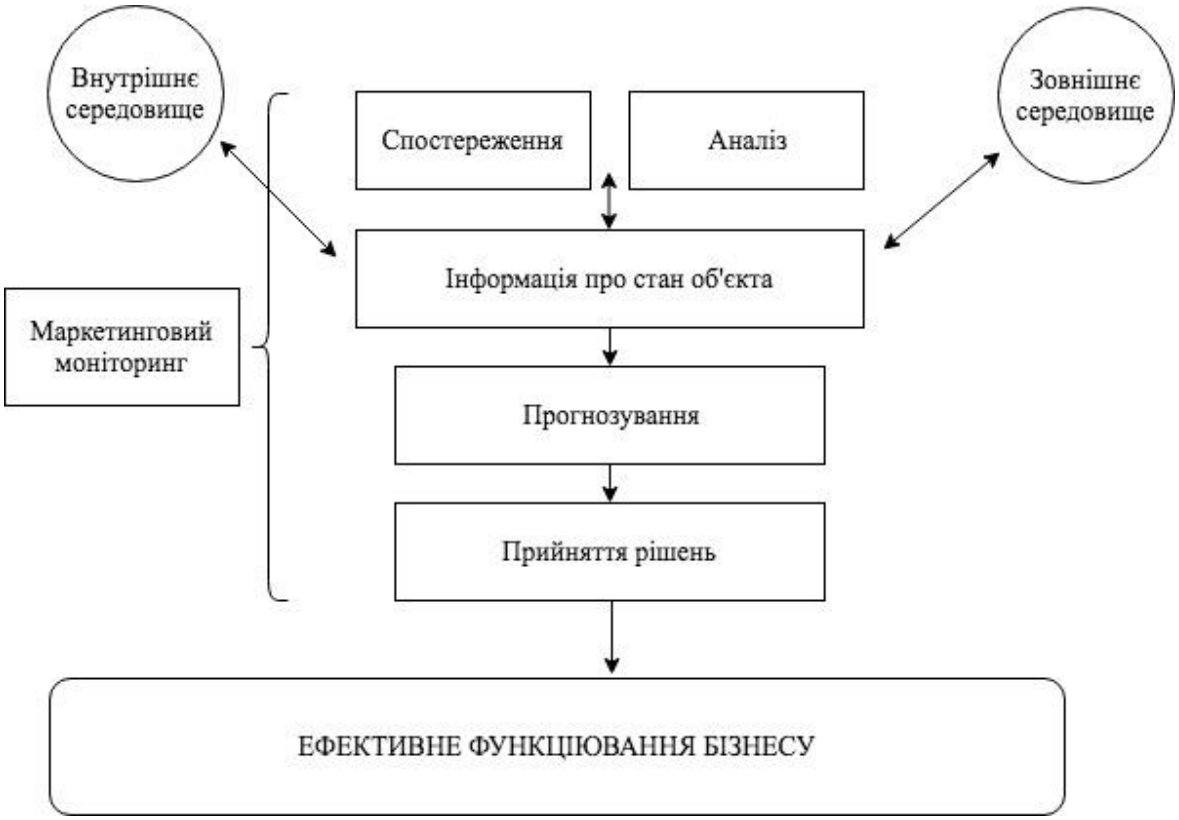

Рис. 1.1. Схема маркетингового моніторингу

В рамках інтегрального підходу маркетинговий моніторинг - це комплексна система спостереження і аналізу зміни показників діяльності підприємства і ринкового середовища його функціонування, що дозволяє прогнозувати і змінювати параметри ефективності його діяльності в перспективі в цілях підвищення якості прийняття обґрунтованих рішень по управлінню.

Завдання і цілі маркетингового моніторингу в системі маркетингового аналізу ринку на основі інтегрального підходу відображені на (Рис. 1.2).

Для реалізації завдань моніторингу важлива роль відводиться забезпеченню високої якості інструментарію, розробці критеріїв оцінювання, індикаторів і показників, процесу вимірювання, статистичній обробці результатів і їх адекватній інтерпретації.

Характеристики моніторингу в системі маркетингового аналізу ринку в контексті інтегрального підходу дозволяють йому виконувати чотири основні функції: передбачити, виявляти, спостерігати, вивчати.

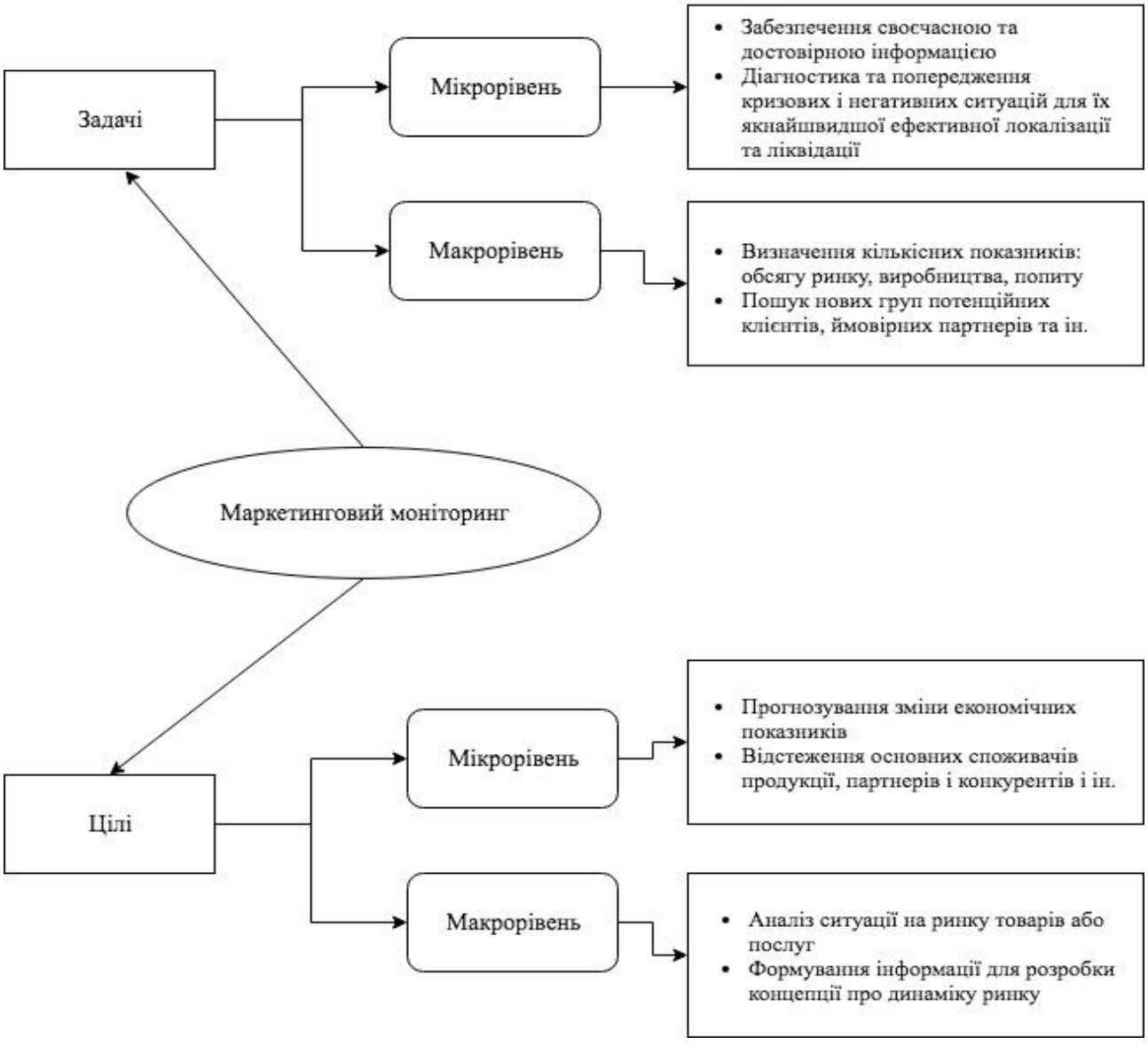

Рис. 1.2. Цілі і задачі маркетингового моніторингу

1.2 Структура системи моніторингу ринку

Структуру системи маркетингового моніторингу ринку можна розбити на три основні частини: початкова інформація, аналітичний розділ і результати досліджень. Відповідно, процес моніторингу ринку складається з трьох основних стадій: пошук інформації, обробка інформації та перетворення її в готовий продукт, представлення результатів дослідження. Усередині кожної із цих стадій існує безліч варіантів, але базова структура процесу досить універсальна (Рис. 1.3).

Рис. 1.3. Основні етапи маркетингового моніторингу ринку

Для кожної з трьох стадій процесу моніторингу ринку існує безліч способів впровадження. При організації даного процесу можна домогтися гарних результатів шляхом планування цих трьох стадій по черзі. Набір

параметрів може сильно варіюватися в залежності від розміру компанії, галузі та багатьох інших факторів. В процесі моніторингу ринку основне завдання полягає в оптимальній комбінації як технологій (для підвищення рівня ефективності та продуктивності процесу), так і людських ресурсів (для генерації ідей), оскільки деякі завдання не можна повністю автоматизувати - оцінка і пошук джерел нової інформації вимагають участі людини, як більшість завдань на стадії обробки.

Залежно від підстав використання можна виділити кілька видів маркетингового моніторингу. Однак найбільш прийнятним може бути підхід до видів моніторингу, в рамках якого виділяють контекстуальний моніторинг (моніторинг середовища) і моніторинг ринкової стратегії.

Перевага використання моніторингу стратегії і прийняття рішень, що ґрунтуються на його результатах, полягає в тому, що організація може підвищити свої можливості щодо виконання робіт шляхом аналізу внутрішньої діяльності по відношенню до контексту (навколишньому середовищу), в якому вона здійснюється, і реалізуючи стратегії, які вона хотіла б застосувати. Маркетинговий моніторинг стратегії концентрує увагу на визначенні чітких зв'язків між такими аспектами, як мета і процес [1].

При наявності прогнозної системи ринкового моніторингу організація може передбачити подію, вжити відповідних заходів і розподілити ресурси, випереджаючи як саму подію, так і конкурентів. Однак варто взяти до уваги таке застереження: яким би прогнозними потенціалом не володіла система моніторингу, якщо вона не доносить отримані відомості до керівництва компанії або якщо це керівництво не готове засновувати на цих відомостях свої рішення, потрібних заходів не буде прийнято навіть в разі прогнозування подій.

Таким чином, система маркетингового моніторингу ринку - це процес вивчення конкурентного середовища і надання детальної інформації особам, які

заміщають різні посади в різних підрозділах компанії, це інструмент, що допомагає реалізовувати поточну стратегію компанії, так як саме вона визначає, яке конкурентне середовище потрібно вивчати. Маркетингова активність компаній на етапі уповільнення зростання ринку досягає свого піку. Для підтримки конкурентних переваг компанії в цей період потрібна ефективна система моніторингу маркетингової діяльності конкурентів, яка дозволяє своєчасно приймати ефективні рішення в області товарної, цінової та комунікативної політики організації.

Конкурентна перевага і конкурентоспроможність - тісно взаємопов'язані і взаємодоповнюючі поняття, сутнісні характеристики яких полягають у тому, що конкурентні переваги є факторною умовою конкуренції, а конкурентоспроможність - рівень успіху, досягнутий в конкурентній боротьбі.

Вивчення впливу різних чинників конкурентоспроможності передбачає виявлення сильних і слабких сторін, можливостей і загроз розвитку, які надають зовнішні та внутрішні, мікро- і макроекономічні чинники на діяльність підприємства.

Критичний аналіз різних точок зору вчених-економістів, які займаються дослідженням проблеми пошуку чинників формування конкурентних переваг, дозволив визначити, що в процесі становлення сучасної теорії конкурентних переваг склалися два підходи: 1) ринковий, де в пошуку джерел конкурентних переваг пріоритет віддається зовнішнім факторам; 2) ресурсний (розвивається школою ресурсів, здібностей і компетенцій), де увагу акцентують на внутрішніх параметрах організації, дослідженні можливостей, потенціалу підприємства.

Завдання моніторингу полягає в діагностиці внутрішнього потенціалу і зовнішніх чинників для забезпечення конкурентоспроможності підприємства. Моніторинг факторів зовнішнього середовища підприємства здійснюється на

першому рівні системи моніторингу, що дозволяє отримувати інформацію про потенційні витрати на утримання і обслуговування всіх груп показників економічного стану. Другий рівень призначений для моніторингу внутрішнього потенціалу підприємства, де оцінюються конкретні виробничі показники, а також тенденції розвитку основних показників економічного стану підприємства.

1.3 Необхідність моніторингу ринку для управління бізнесом

Постійну конкурентоспроможність потенціалу організації, економічний і соціальний попит товару на ринку забезпечує його невід'ємна частина - маркетинговий потенціал, який є інтегральною характеристикою ступеня використання маркетингового ресурсного забезпечення підприємства.

Таким чином, моніторинг ринку необхідний для ефективного управління бізнесом. Він дозволяє відстежувати діяльність прямих конкурентів і їх цінову політику. Виходячи з цих параметрів бізнес можна підлаштовувати під основні тенденції аналізованого ринку з метою збільшення клієнтської бази і відповідно рентабельності бізнесу. Результати моніторингу дають можливість вносити коригування в політику маркетингу та управління [2].

У сучасному діловому світі більшість великих компаній мають власні системи моніторингу ринку, що дозволяють їм збирати інформацію про конкурентів, клієнтів та інших учасників ринку. Система маркетингового моніторингу ринку сприяє підвищенню конкурентоспроможності в тому випадку, якщо вона дозволяє прогнозувати майбутні зміни ділового середовища.

Управління на основі моніторингу зовнішнього середовища і внутрішніх можливостей підприємства дозволить, по-перше, забезпечити його ефективне функціонування в умовах турбулентного середовища, а по-друге, активно впливати на формування і розвиток ринкового середовища.

І внутрішнє середовище, і зовнішнє оточення вивчаються стратегічним управлінням в першу чергу для того, щоб розкрити ті загрози і можливості, які компанія повинна враховувати при постановці і реалізації своїх цілей.

Основна перевага прогнозної спрямованості системи моніторингу ринку полягає в тому, що вона дозволяє керівництву компанії не реагувати на події, а передбачати їх. Більш того, передбачення передбачає два аспекти: гру на випередження ринку (вжиття заходів до настання події) і гру на випередження конкурентів. В ідеалі компанія повинна прагнути робити і те й інше.

Використання системи моніторингу протягом певного періоду дозволить компанії більш виважено позиціонувати себе серед організацій-конкурентів, а також здійснювати аналіз змін не тільки свого економічного стану, а й галузевих тенденцій. Це дасть можливість розробляти ефективну стратегію компанії, коригувати інвестиційні плани.

Складність створення системи управлінського моніторингу полягає в тому, що не можна підходити до моніторингу догматично, на основі наявних схем, без належного усвідомлення і осмислення його сутності, без урахування особливостей конкретної соціальної системи. Система моніторингу підприємства повинна практично в режимі реального часу проводити незалежні оцінки тенденцій розвитку його економічного стану, отримувати інформацію про стан економічної кон'юнктури в реальному секторі економіки і її можливі зміни, оперативно проводити аналіз фінансового стану підприємства і найважливіших факторів, що визначають його інноваційно-інвестиційну активність.

При організації системи моніторингу ринку в першу чергу слід розуміти різницю між реалізацією і формулюванням стратегії, так як ці два процеси висувають різні вимоги до процесу моніторингу ринку.

Модель підвищення конкурентоспроможності компанії на основі маркетингового моніторингу ринку полягає у проведенні на першому етапі моніторингу, який буде складатися з наступних стадій: аналіз поточного стану компанії і факторів, що впливають на її розвиток; пошук кращої практики серед конкурентів, який в подальшому перетече в визначення місць, які зменшують ефективність діяльності компанії, т. е. точок розривів. Далі необхідно провести діагностику бізнес-процесів і оцінити варіанти їх поліпшення за показниками якості, результативності та ефективності. Дослідження, проведені в зазначених напрямках, дадуть компанії можливість визначити конкурентний клімат, інтенсивність суперництва і тип конкурентної переваги, яким володіє кожен з конкурентів. Дані дії будуть сприяти формуванню конкурентної моделі розвитку компанії, яка дозволить розробити бізнес-план і залучати інвесторів. Крім того, на основі даної моделі доцільно здійснювати розробку інноваційної стратегії підприємства.

Підвищення конкурентоспроможності по представленій моделі відбувається на основі максимально ефективного використання внутрішнього соціально-економічного потенціалу компанії і приведення його у відповідність до вимог зовнішнього середовища, ринку.

Таким чином, маркетинговий моніторинг є засобом трансформації соціально-економічного потенціалу (джерел конкурентних переваг) в конкурентні переваги і конкурентоспроможність компанії в цілому в результаті реалізації інноваційної стратегії розвитку бізнесу і застосування проактивної моделі управління компанією.

Особливість системи маркетингового моніторингу ринку, спрямованої на підвищення конкурентоспроможності підприємства, полягає в створенні системи регулярних відстежень, що дозволяє виявити динаміку процесів. Тому планування повинно спиратися на аналіз результатів моніторингу процесів розвитку ринку.

Згідно з розробленою системою на кожному етапі маркетингового моніторингу здійснюються певні операції, супроводжувані передачею інформації по каналах зв'язку. Створюється система постійної циркуляції інформаційних потоків, в якій рух даних носить циклічний характер, що передбачає набір повторюваних дій з боку учасників моніторингу. При розробці системи інформаційних комунікацій в рамках конкретної компанії потрібен індивідуальний підхід з урахуванням специфіки наявних усталених взаємозв'язків в компанії.

Стратегічною метою створення системи моніторингу діяльності конкурентів є інформаційне забезпечення процесу формування конкурентних переваг. Проведений моніторинг забезпечує порівняльний аналіз стану і динаміки розвитку підприємства, виявлення трансформаційних процесів.

При плануванні маркетингового моніторингу ринку фахівцям компанії слід пам'ятати, що ефект від його проведення повинен бути істотно вище витрат на нього. Маркетинговий моніторинг не повинен перетворюватися на самоціль, це лише інструмент в удосконаленні системи маркетингу. Методи маркетингового моніторингу ринку повинні відбиратися за принципом оптимальності витрат і ефективності (якості) моніторингу [3].

Посилення уваги до маркетингового моніторингу випливає з необхідності враховувати швидкість трансформації зовнішнього середовища, в тому числі положення про суперечливий, але взаємопов'язаний характер цих змін.

Здатність заздалегідь виявити свої можливості при виникненні нової тенденції споживання, зміну законодавчої бази або появі нових технологій важлива сама по собі. Однак здатність зробити це раніше за конкурентів означає можливість захопити більшу частку ринку, отримати більше прибутку або підвищити імідж бренду, іншими словами, підвищити

конкурентоспроможність підприємства. Це все допомагає зробити система маркетингового моніторингу.

Таким чином, моніторинг розглядається нами як сучасна система, що дозволяє ефективно діагностувати та оцінити внутрішній потенціал компанії і за допомогою оперативного відстеження змін зовнішнього середовища створювати сприятливі умови для його ефективного використання та підвищення конкурентоспроможності компанії. В результаті моніторинг постає засобом трансформації соціально-економічного потенціалу (джерел конкурентних переваг) в конкурентні переваги і конкурентоспроможність компанії в цілому.

Здійснюючи маркетинговий моніторинг, компанії отримують можливість забезпечити себе об'єктивною інформацією про себе і конкурента, провести порівняння, правильно оцінити ситуацію на ринку і спрогнозувати її розвиток, а отже, отримати конкурентні переваги і тим самим знизити рівень комерційного ризику, знайти для себе відповідний сегмент ринку і ринкову нішу, вибрати правильний напрямок диверсифікації, встановити оптимальний рівень цін. Тому вивчення стану ринку є необхідною умовою підприємництва.

Висновок до розділу 1

Сьогодні маркетинговий моніторинг украй затребуваний в компаніях, які активно позиціонують себе на ринку. Актуальна, точна і добре структурована інформація про конкурентів виступає одним з ключових факторів успішного просування. Отже, система моніторингу ринку повинна не просто вирішувати питання накопичення даних і видачі звітів. Головне її завдання - забезпечення осіб, котрі приймають управлінські рішення, релевантною інформацією, яка допомагає вибрати оптимальний варіант вирішення проблеми, що стоїть перед компанією і простежити реакцію конкурентів на прийняте рішення. Для

отримання надійних висновків моніторинг компанії повинен здійснюватися з дотриманням основоположних принципів. Він не повинен бути ізольований від інших бізнес-процесів.

Основоположні принципи моніторингу:

- спостереження за станом ринку;
- обробка і аналіз інформації;
- аналіз і оцінка стану ринку;
- контроль за станом ринку;
- прогнозування стану ринку;
- попередження про можливе настання несприятливих ситуацій;
- ухвалення управлінських рішень.

2. МАШИННЕ НАВЧАННЯ

2.1 Основні поняття і позначення

Машинне навчання - це підрозділ штучного інтелекту, в якому вивчаються алгоритми, здатні навчатися без прямого програмування того, що потрібно вивчати. Машинне навчання широко застосовується в багатьох областях, які, так чи інакше, займаються збором і аналізом даних. Машинне навчання є, по суті, розвитком методів апроксимації функції, але в якості точок виступають більш складні об'єкти, елементи складно-описаних просторів, а відповідями можуть бути не тільки числа, а й множини.

Завдання машинного навчання зводяться до задачі знаходження невідомої залежності між відомою безліччю об'єктів і безліччю відповідей. Тобто необхідно побудувати таку функцію, яка б достатньо точно наближала значення безлічі відповідей в точках безлічі об'єктів і на всьому іншому просторі.

2.1.1 Об’єкти і відповіді

Об'єктом може бути будь-що: веб-сторінки, країни, люди, вироби, фірми - усе, що несе певну інформацію (має набір ознак). Під ознаками розуміються способи вимірювання характеристик об'єктів в досліджуваному просторі. Залежно від безлічі допустимих значень всі ознаки можуть бути розділені на:

- бінарні;
- номінальні;
- порядкові;
- кількісні;
- якісні.

Залежно від відповідей (значень цільової змінної), завдання машинного навчання поділяються на типи. Основні типи завдань машинного навчання:

- завдання класифікації (множина класів відповідей скінченна: 2 класи, M пересічних класів, M непересічних класів);
- завдання регресії (прогнозування речової величини або вектора таких величин);
- завдання ранжирування (множина впорядковується за заданою ознакою).

2.1.2 Модель алгоритмів

Модель алгоритмів (модель залежності) - параметричне сімейство відображень, в якому потрібно знайти функцію, котра наближає цільову залежність, функціональну або стохастичну залежність між об'єктами і відповідями. Завдання знаходження моделі залежності зводиться до побудови алгоритму, який би однаково точно наближав невідому цільову залежність, як на елементах вибірки, так і на всьому просторі об'єктів. Це завдання отримало назву - навчання з учителем (supervised learning).

Як правило, залежність параметрів будується виходячи із законів досліджуваної області, наприклад фізична модель. У найпростішому випадку можна використовувати лінійну модель.

2.1.3 Метод навчання

Метод навчання - відображення, яке ставить у відповідність навчальній вибірці алгоритм із заданої моделі. Іншими словами, алгоритм будується по вибірці.

Процес навчання поділяється на два етапи:

- Етап навчання - процес виявлення залежностей в емпіричних даних.
- Етап застосування - процес оцінки точності, виявленої залежності.

На етапі навчання для виявлення залежності використовується навчальна вибірка (training sample), по ній проводиться оптимізація параметрів. Для

оцінки якості моделі використовується тестова, або контрольна, (test sample) вибірка. Щоб уникнути зсуву оцінки, навчальна і тестова вибірки повинні бути незалежними. Для вибору найкращої моделі з безлічі моделей, побудованих на навчальній вибірці, використовується перевірочна вибірка (validation sample). Важливою умовою є те, щоб навчальна вибірка повинна володіти достатньою повнотою, тобто повинна покривати всі можливі випадки.

2.1.4 Функціонали якості

Для оцінки алгоритму, а точніше для його застосування на вибірці, слід, перш за все, оцінити застосування алгоритму на окремому об'єкті. Для формалізації поняття величини помилки алгоритму на об'єкті вводиться поняття - функція втрат. Функція втрат характеризує величину відхилення відповіді моделі від правильної відповіді на довільному об'єкті.

Емпіричний ризик - функціонал алгоритму на навчальній вибірці, характеризує якість наближення заданої функції на вибірці і дорівнює середньому значенню втрат (сума значень функції втрат за навчальною вибіркою, поділена на довжину вибірки). Іншими словами, емпіричний ризик - середня величина помилки алгоритму на навчальній вибірці. Таким чином, завдання навчання зводяться до задач оптимізації, знаходження функцій наближення з мінімальним значенням функції втрат, тобто зводити завдання до чисельних методів оптимізації [4].

Навчальна (узагальнююча) здатність алгоритму характеризується співвідношенням точності наближень на тестовій і навчальної вибірках. Якщо ймовірність виникнення помилок на тестовій вибірці мала і близька до значення ймовірності помилок на навчальній вибірці, то можна говорити про хорошу навчальної спроможності алгоритму, інакше ми стикаємося з такими явищами, як недонавчання і перенавчання.

Недонавчання виникає при використанні недостатньо складних моделей і характеризується великою величиною помилки на навчальній вибірці.

Перенавчання виникає при використанні надмірно складних моделей. Перенавчанням називають стан, коли ймовірність помилки на об'єктах тестової вибірки істотно вище ймовірності на навчальній.

Перенавчання виникає через надмірну складність простору параметрів, зайві ступені свободи в моделі «витрачаються» на надмірно точну підгонку під навчальну вибірку. Перенавчання є завжди, коли є оптимізація параметрів по кінцевій (завідомо неповній) вибірці.

Існують різні методи виявлення перенавчання. Емпірично перенавчання можна виявити за допомогою вибіркового контролю.

Зовсім позбутися від перенавчання не можна, можна його лише мінімізувати. Один із способів - введення обмежень на простір параметрів.

2.2 Огляд алгоритмів машинного навчання

Немає такого алгоритму, який був би кращим вибором для кожного завдання, що особливо стосується навчання з учителем. Наприклад, не можна сказати, що нейронні мережі завжди працюють краще, ніж дерева рішень, і навпаки. На ефективність алгоритмів впливає безліч факторів на кшталт розміру і структури набору даних.

З цієї причини доводиться пробувати багато різних алгоритмів, перевіряючи ефективність кожного на тестовому наборі даних, і потім вибирати кращий варіант. Алгоритми машинного навчання можна описати як навчання цільової функції f, яка найкращим чином співвідносить вхідні змінні X і вихідну змінну Y: $Y = f(X)$.

2.2.1 Лінійна регресія

Лінійна регресія - один з найбільш відомих і зрозумілих алгоритмів в статистиці і машинному навчанні. Прогностичне моделювання в першу чергу стосується мінімізації помилки моделі або, іншими словами, як можна більш точного прогнозування. Алгоритми запозичуються з різних областей, включаючи статистику.

Лінійну регресію можна представити у вигляді рівняння, яке описує пряму, що найбільш точно показує взаємозв'язок між вхідними змінними X і вихідними змінними Y. Для складання цього рівняння потрібно знайти певні коефіцієнти B для вхідних змінних.

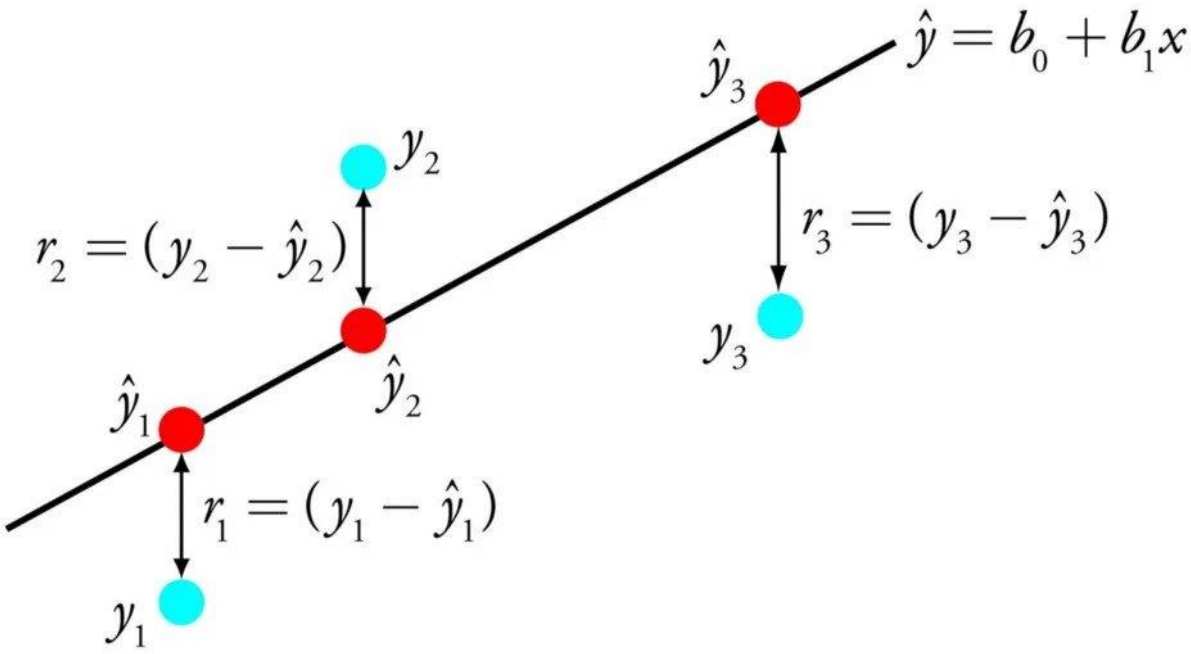

Рис. 2.1. Лінійна регресія

Наприклад: Y = B0 + B1 * X. Знаючи X, ми повинні знайти Y, і мета лінійної регресії полягає в пошуку значень коефіцієнтів B0 і B1. Для оцінки регресійній моделі використовуються різні методи на кшталт лінійної алгебри або методу найменших квадратів.

2.2.2 Логістична регресія

Логістична регресія - алгоритм, який прийшов в машинне навчання з статистики. Використовується для завдань бінарної класифікації (це завдання, в яких на виході ми отримуємо один з двох класів). Логістична регресія схожа на лінійну тим, що в ній теж потрібно знайти значення коефіцієнтів для вхідних змінних. Різниця полягає в тому, що вихідне значення перетвориться за допомогою нелінійної або логістичної функції.

Логістична функція виглядає як велика буква S і перетворює будь-яке значення в число в межах від 0 до 1.

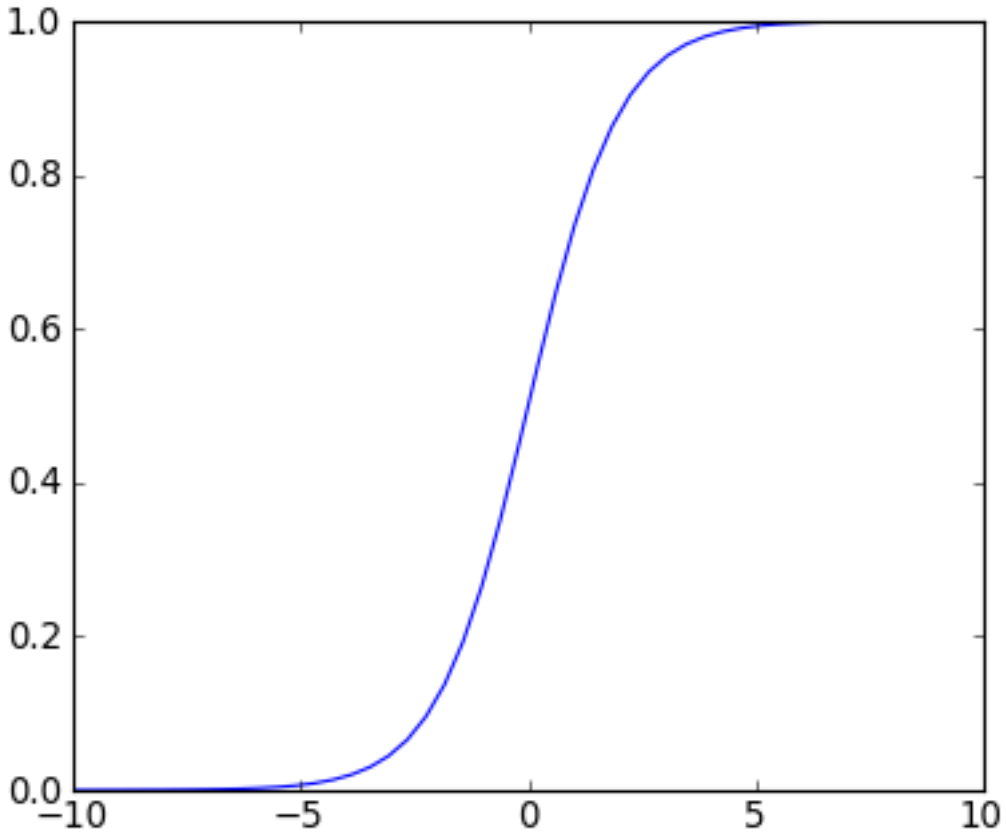

Рис. 2.2. Логістична регресія

Завдяки тому, як навчається модель, передбачення логістичної регресії можна використовувати для відображення ймовірності приналежності зразка до класу 0 або 1. Це корисно в тих випадках, коли потрібно мати більше обґрунтувань для прогнозування. Як і у випадку з лінійної регресією,

логістична регресія виконує своє завдання краще, якщо прибрати зайві і схожі змінні. Модель логістичної регресії швидко навчається і добре підходить для задач бінарної класифікації.

2.2.3 Лінійний дискримінантний аналіз (LDA)

Логістична регресія використовується, коли потрібно віднести зразок до одного з двох класів. Якщо класів більше, ніж два, то краще використовувати алгоритм LDA (Linear discriminant analysis).

Подання LDA досить просте. Воно складається зі статистичних властивостей даних, розрахованих для кожного класу. Для кожної вхідної змінної це включає:

- середнє значення для кожного класу;
- дисперсію, розраховану по всіх класах.

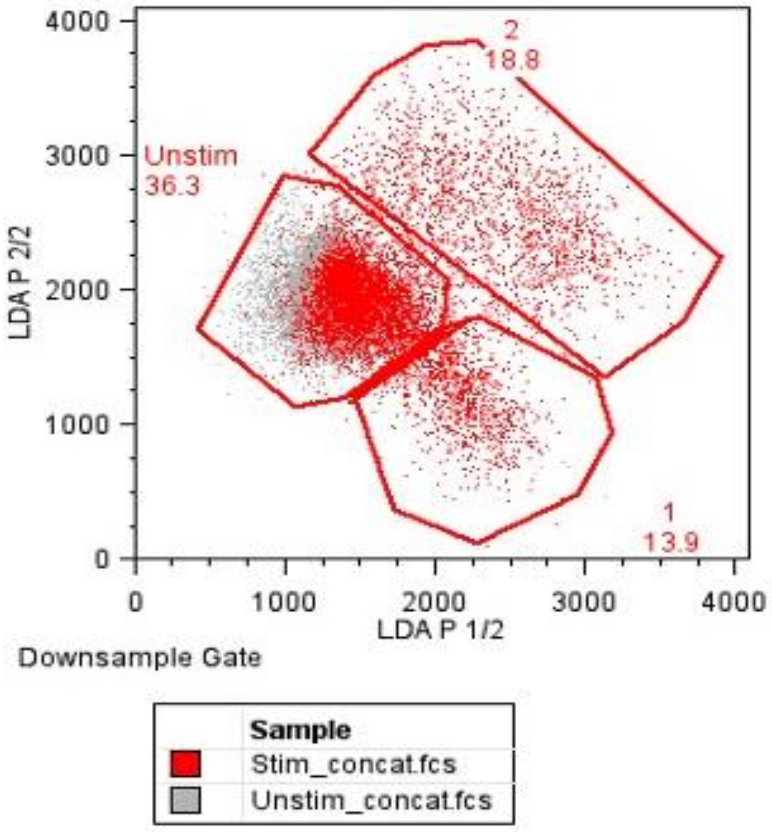

Рис. 2.3. Лінійний дискримінантний аналіз

Передбачення виробляються шляхом обчислення дискримінантного значення для кожного класу і вибору класу з найбільшим значенням. Передбачається, що дані мають нормальний розподіл, тому перед початком роботи рекомендується видалити з даних аномальні значення.

2.2.4 Дерева прийняття рішень

Дерево рішень можна представити у вигляді двійкового дерева, знайомого багатьом по алгоритмам і структурам даних. Кожен вузол являє собою вхідну змінну і точку поділу для цієї змінної (за умови, що змінна - число).

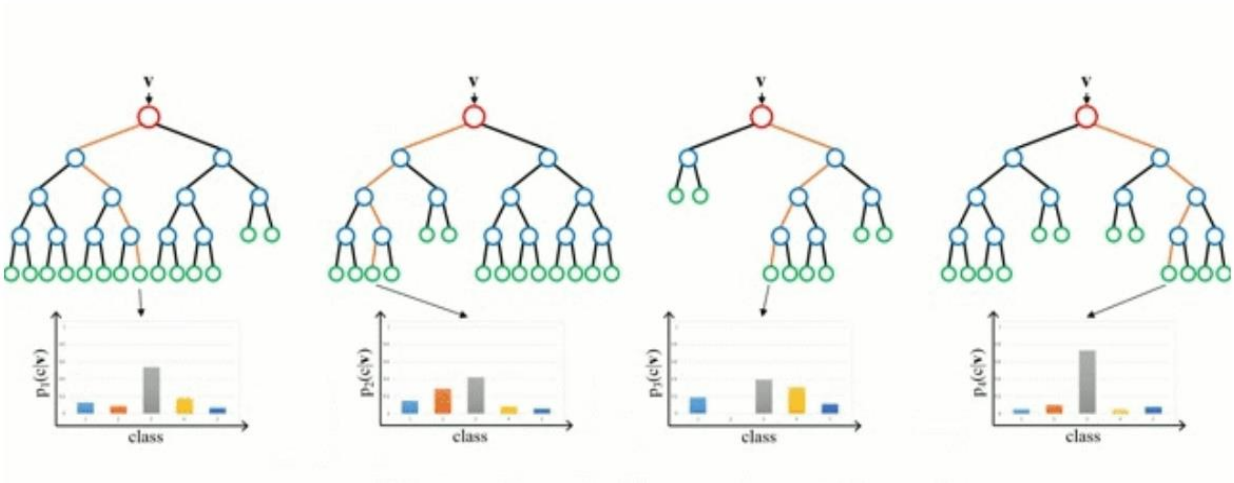

Рис. 2.4. Дерево прийняття рішень

Листові вузли - це вихідна змінна, яка використовується для передбачення. Передбачення виробляються шляхом проходу по дереву до листового вузлу і виведення значення класу на цьому вузлі.

Дерева швидко навчаються і роблять прогнози. Крім того, вони точні для широкого кола завдань і не вимагають особливої підготовки даних.

2.2.5 Наївний Байєсівський класифікатор

Простий, але ефективний алгоритм, модель якого складається з двох типів ймовірностей, які розраховуються за допомогою тренувальних даних:

- імовірність кожного класу;
- умовна ймовірність для кожного класу при кожному значенні x.

Після розрахунку імовірнісної моделі її можна використовувати для передбачення з новими даними за допомогою теореми Байеса. Якщо у вас речові дані, то, припускаючи нормальний розподіл, розрахувати ці ймовірності не складає особливої складності.

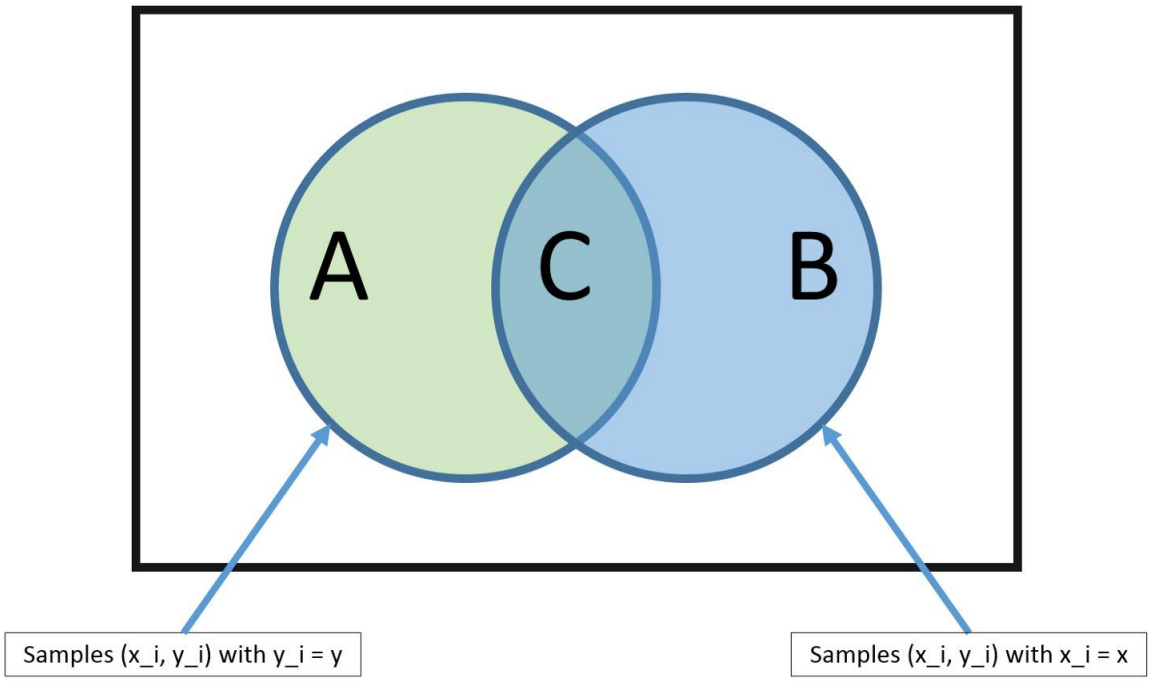

Рис. 2.5. Наївний Байєсівський класифікатор

Наївний Байєсівський класифікатор називається наївним, тому що алгоритм передбачає, що кожна вхідна змінна незалежна. Це сильне припущення, яке не відповідає реальним даним. Проте даний алгоритм дуже ефективний для цілого ряду складних завдань на зразок класифікації спаму або розпізнавання рукописних цифр.

2.2.6 Нейронні мережі

Нейронна мережа складається з взаємозв'язаних груп вузлів, званих нейронами. Вхідні дані передаються в ці нейрони в вигляді лінійної комбінації

з безліччю змінних. Значення, яке множиться на кожну функціональну змінну, називається вагою. Потім до цієї лінійної комбінації застосовується нелінійність, що дає нейронній мережі можливість моделювати складні нелінійні відносини. Найчастіше нейронні мережі бувають багатошаровими: вихід одного шару передається наступному так, як описано вище. На виході нелінійність не застосовується.

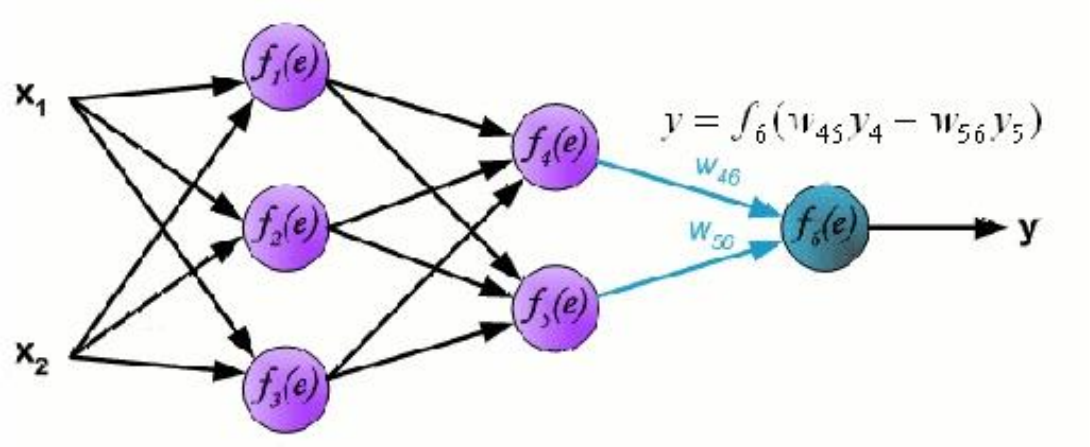

Рис. 2.6. Нейронна мережа

Нейронні мережі тренуються за допомогою методу стохастичного градієнта і алгоритму зворотного поширення помилки.

2.2.7 K-найближчих сусідів (KNN)

K-найближчих сусідів - дуже простий і дуже ефективний алгоритм. Передбачення для нової точки робиться шляхом пошуку K найближчих сусідів в наборі даних і підсумовування вихідної змінної для цих K примірників.

KNN може потребувати багато пам'яті для зберігання всіх даних, але здатний швидко зробити прогноз. Також навчальні дані можна оновлювати, щоб передбачення залишалися точними з плином часу.

Ідея найближчих сусідів може погано працювати з багатовимірними даними (безліч вхідних змінних), що негативно позначиться на ефективності алгоритму при вирішенні задачі. Це називається проблемою розмірності.

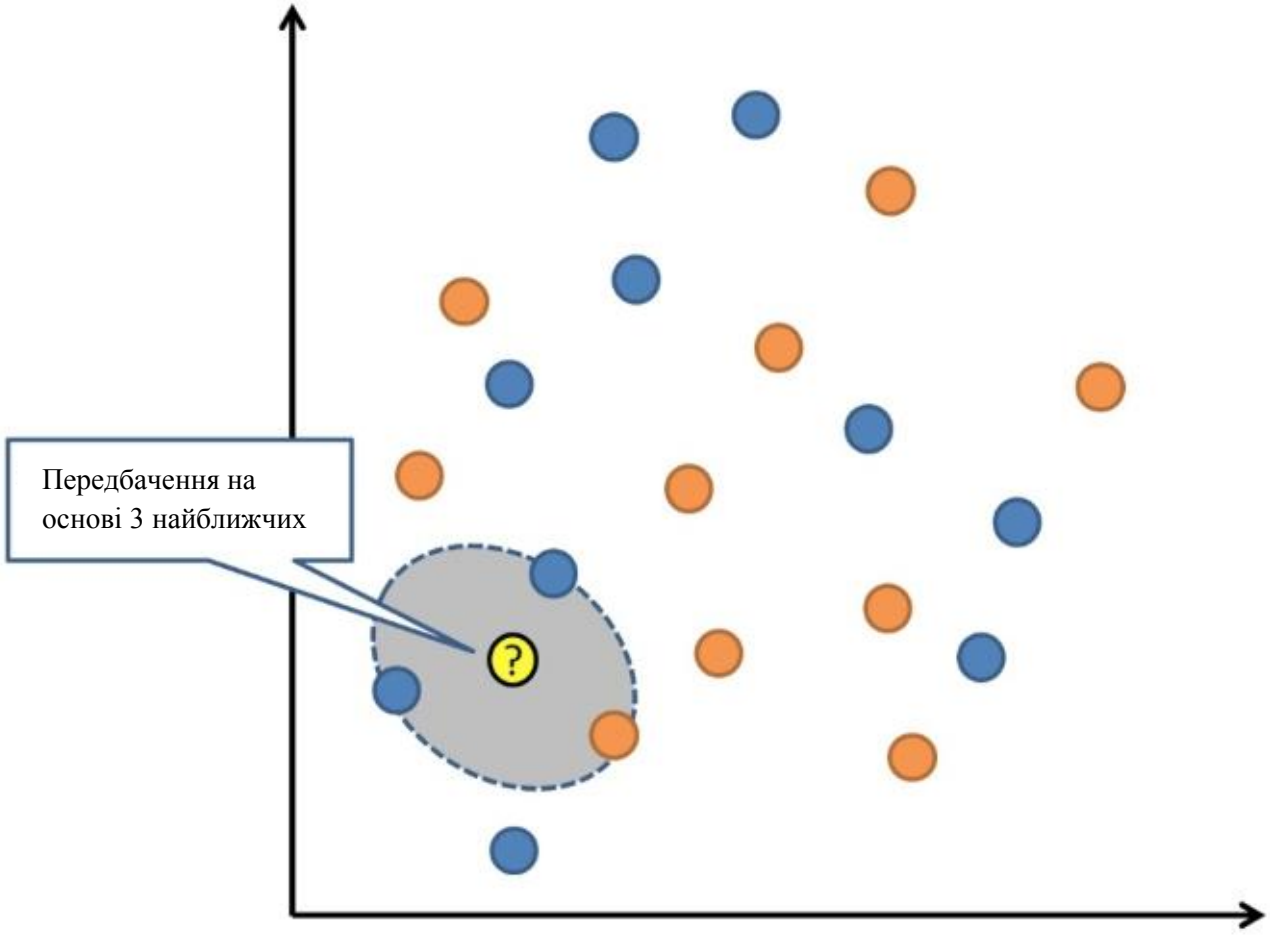

Рис. 2.7. К-найближчих сусідів

Іншими словами, варто використовувати лише найбільш важливі для передбачення змінні.

2.2.8 Метод опорних векторів (SVM)

Метод опорних векторів, один з найбільш популярних і обговорюваних алгоритмів машинного навчання.

Гіперплощина - це лінія, що розділяє простір вхідних змінних. У методі опорних векторів гіперплощина вибирається так, щоб найкращим чином розділяти точки в площині вхідних змінних по їх класу: 0 або 1. В двовимірній площині це можна уявити як лінію, яка повністю поділяє точки всіх класів. Під час навчання алгоритм шукає коефіцієнти, які допомагають краще розділяти класи гіперплощини [5].

Відстань між гіперплощиною і найближчими точками даних називається різницею. Краща або оптимальна гіперплощина, що розділяє два класи, - це лінія з найбільшою різницею. Тільки ці точки мають значення при визначенні гіперплощини і при побудові класифікатора. Ці точки називаються опорними векторами. Для визначення значень коефіцієнтів, що максимізують різницю, використовуються спеціальні алгоритми оптимізації.

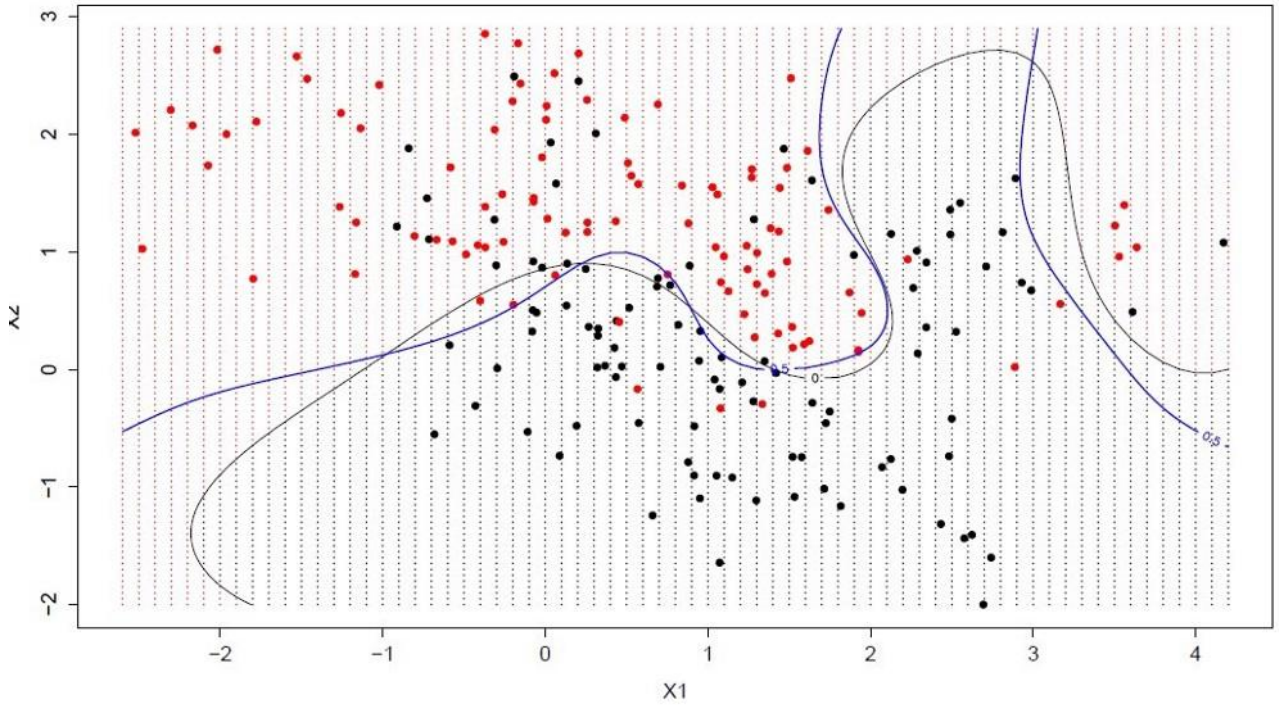

Рис. 2.8. Метод опорних векторів

Метод опорних векторів, один з найефективніших класичних класифікаторів, на який безперечно варто звернути увагу.

2.2.9 Випадковий ліс

Випадковий ліс - популярний і ефективний алгоритм машинного навчання, що являє собою сукупність дерев прийняття рішень. Вхідний вектор проходить через кілька дерев рішень. Для регресії вихідні значення всіх дерев усереднюються; для класифікації використовується схема голосування для визначення кінцевого класу. Це різновид ансамблевого алгоритму, званого беггінгом.

У беггінгу для оцінки всіх статистичних моделей найчастіше використовуються дерева рішень. Тренувальні дані розбиваються на безліч вибірок, для кожної з яких створюється модель. Коли потрібно зробити прогноз, то його робить кожна модель, а потім передбачення усереднюються, щоб дати кращу оцінку значенню.

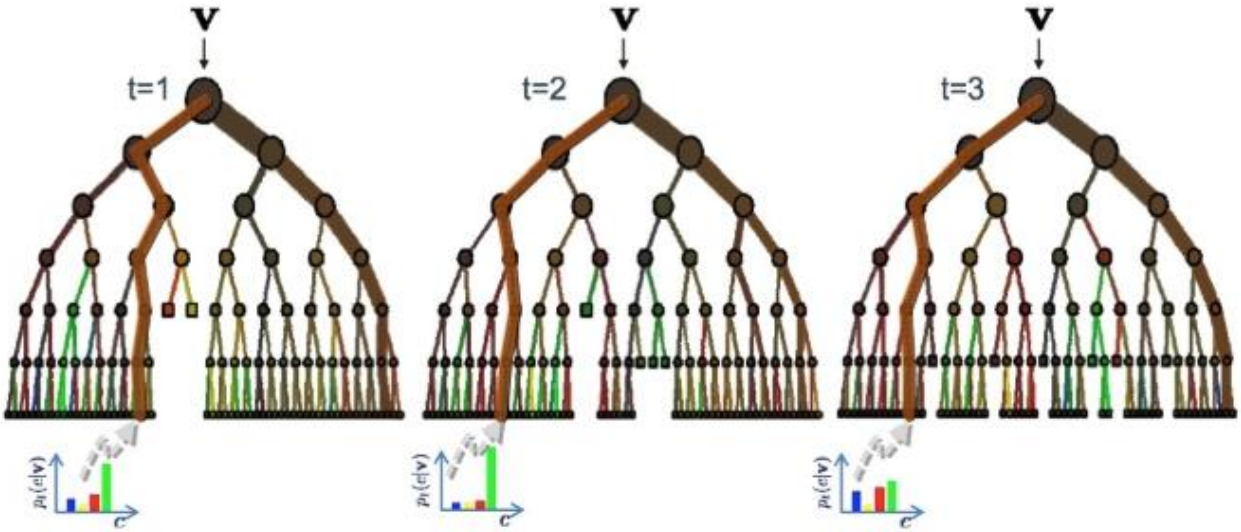

Рис. 2.9. Випадковий ліс

В алгоритмі випадкового лісу для всіх вибірок з тренувальних даних будуються дерева рішень. При побудові дерев для створення кожного вузла вибираються випадкові ознаки. Окремо отримані моделі не дуже точні, але при їх об'єднанні якість передбачення значно поліпшується. Використовувати великі випадкові ліси для досягнення більш високої продуктивності, невигідно, оскільки ідуть великі витрати пам'яті і часу.

2.3 Галузі застосування методів машинного навчання

Більшість галузей, що працюють з великими обсягами даних, визнали цінність технологій машинного навчання. Побудова точних моделей допомагає виявляти вирішальні чинники і отримувати корисну інформацію, яка може допомогти підприємствам збільшити прибуток і уникнути ризикованих угод, що підвищує ефективність їх роботи і дає їм перевагу перед конкурентами.

Алгоритми машинного навчання уже довгий час оточують нас. Ось кілька широко відомих прикладів машинного навчання, про які чув кожен:

- функціонування самокерованого автомобіля Google засноване на методах машинного навчання;
- онлайн рекомендації таких сервісів як Amazon і Netflix є яскравими прикладами машинного навчання для повсякденного життя;
- можливість дізнатися, що інші говорять про вас в Твіттері - сукупність машинного навчання і лінгвістичних правил.
- боротьба з шахрайством, одне з найбільш очевидних і важливих напрямків сьогодні.

Машинне навчання вже давно широко використовується в галузі фінансових послуг, урядових органах, в охороні здоров'я та маркетингу.

Банки та інші підприємства фінансової індустрії використовують машинне навчання для двох основних цілей: виявлення взаємозв'язків в даних і запобігання шахрайства. Отримані знання можуть допомогти виявити інвестиційні ризики, що проінформує інвесторів про найбільш вигідні угоди. Інтелектуальний аналіз даних сприяє ідентифікації неплатоспроможних клієнтів, а також може використовуватися як інструмент кібер-спостереження для виявлення шахрайства.

Урядові установи, такі як служба безпеки і комунальні послуги працюють з декількома різнорідними джерелами даних, тому застосування методів машинного навчання є актуальним і вирішує такі важливі завдання як виявлення шахрайства та ліквідація його наслідків з найменшими втратами.

Розвинуте застосування методів машинного навчання в галузі охорони здоров'я обумовлено величезною кількістю різних пристроїв і датчиків, які допомагають відслідковувати стан пацієнта в режимі реального часу, тобто генерують великий обсяг даних. Ці дані аналізуються за допомогою методів

машинного навчання, що допомагає лікарям у виявленні важливих факторів і в постановці діагнозу.

Висновок до розділу 2

В даному розділі дипломного проекту було розглянуто машинне навчання, алгоритми які в ньому використовуються та галузі застосування методів машинного навчання. Був проведений огляд декількох алгоритмів машинного навчання: лінійна регресія, логістична регресія, лінійний дискримінантний аналіз, нейронна мережа, метод опорних векторів, дерево прийняття рішень, наївний Байєсівський класифікатор, k-найближчих сусідів, випадковий ліс.

Машинне навчання являє собою великий підрозділ штучного інтелекту, який використовує розділи математичної статистики, чисельних методів оптимізації, теорії ймовірностей, дискретного аналізу, що виділяє знання з даних. Побудова систем машинного навчання є на сьогоднішній день однією з найпопулярніших, актуальних і сучасних областей людської діяльності на стику інформаційних технологій, математичного аналізу та статистики.

Машинне навчання все глибше проникає в наше життя за допомогою призначених для користувача продуктів, створених за допомогою методів штучного інтелекту. Очевидно, що дані технології будуть розвиватися і далі, поступово стаючи частиною повсякденної рутини в багатьох областях людської професійної діяльності.

3. ВИБІР ІНСТРУМЕНТІВ РЕАЛІЗАЦІЇ

3.1 Мова програмування Python

Python - високорівнева універсальна інтерпретована мова сценаріїв. При розробці мовою Python велика увага приділяється простоті і зрозумілості синтаксису, що не тільки скорочує час вивчення його основ, але і підвищує швидкість розробки в цілому. Це далеко не всі переваги даної мови, основні з них:

- об'єктно-орієнтованість;
- вільне поширення і широка підтримка;
- кросплатформеність;
- розвинені функціональні можливості.

Мова Python об'єднує дві парадигми програмування - об'єктно орієнтовану, яка є потужним засобом структурованого програмного коду для багаторазового використання, і процедурну, що розширює коло вирішення завдань, дозволяючи використовувати Python при рішенні тактичних завдань з відсутністю фази проектування. Об'єктна модель Python підтримує поняття поліморфізму, перевантаження операторів і множинного наслідування.

Кросплатформеність мови досягається завдяки його реалізації на ANSI C, що дозволяє програмам, написаним на мові Python, однаково добре компілюватиметься і виконуватися на будь-яких платформах, де встановлена сумісна версія Python.

Гібридна природа Python об'єднує в собі простоту і зручність мов сценаріїв і потужності компілюємих мов, що робить Python зручним засобом розробки додатків різного типу. Однак найбільша ефективність мови досягається при вирішенні задач аналізу даних і автоматизації процесів. Python широко використовується в дослідницьких роботах. Мова програмування Python має

потужний вбудованим інструментарієм (вбудовані типи об'єктів і динамічна типізація, автоматичне керування пам'яттю) і можливість використання зовнішніх бібліотек і утиліт сторонніх розробників для вирішення більш вузькоспеціалізованих завдань.

Python використовується не тільки окремими користувачами, але і компаніями, включаючи комерційне використання. Наприклад:

- Компанія Google широко використовує Python в своїй пошуковій системі і для створення фреймворка App Engine.
- Служба колективного використання відеоматеріалів YouTube в значній мірі реалізована на мові Python.
- Популярна програма BitTorrent написана на мові Python.
- Такі компанії, як EVE Online і Massively Multiplayer Online Game, широко використовують Python в своїх розробках.
- Потужна система тривимірного моделювання і створення мультиплікації Maya підтримує інтерфейс для управління з сценаріїв на мові Python.
- Виробники електронних пристроїв і комп'ютерних компонентів (такі, як Intel, Cisco, Hewlett-Packard, Seagate, Qualcomm і IBM) використовують Python для тестування апаратного забезпечення.
- Кіностудії, що займаються створенням візуальних ефектів до кінофільмів (Industrial Light & Magic, Pixar і інші) використовують Python у виробництві анімаційних фільмів.
- Найбільші фінансові конгломерати (JPMorgan Chase, Union Bank of Switzerland, Federal Credit Union) застосовують Python для прогнозування фінансового ринку.
- Науково-дослідні центри (NASA, Los Alamos, Fermilab, Jet Propulsion Laboratory і інші) використовують Python для наукових обчислень.
- Компанія з розробки та продажу робото-техніки iRobot використовує Python в розробці комерційних роботизованих пристроїв.

- Компанія з виробництва геоінформаційних систем (Environmental Systems Research Institute) використовує Python в якості інструменту налаштування своїх програмних продуктів під потреби кінцевого користувача.
- Агентство національної безпеки Сполучених штатів (National Security Agency) використовує Python для шифрування і аналізу розвідданих.

3.2 Бібліотека scikit-learn

Завдяки широкому поширенню Python зібрав навколо себе активну спільноту розробників, які в рамках різних проектів розробляють модулі для вузькоспеціалізованих завдань.

Одним з таких проектів став Google Summer of Code 2007, в рамках якого David Cournapeau розробив бібліотеку scikit-learn. Розробка даної бібліотека є однією з причин популяризації застосування мови Python в області аналізу даних за допомогою методів машинного навчання.

Бібліотека scikit-learn надає реалізацію ряду алгоритмів як для навчання з учителем (Supervised learning), так і для навчання без (Unsuoervised learning).

Scikit-learn побудована на основі стека SciPy (Scientific Python), який включає в себе:

- NumPy додає підтримку великих багатовимірних масивів і матриць, а також бібліотеку високорівневих математичних функцій для операцій з ними.
- SciPy - відкрита бібліотека високоякісних наукових інструментів для мови програмування Python.
- Matplotlib - бібліотека для візуалізації двовимірної і тривимірної графіки.
- IPython - інтерактивна оболонка для мови програмування Python, яка надає розширену інтроспекцію, додатковий командний синтаксис,

підсвічування коду і автоматичне доповнення.

- Sympy - бібліотека для роботи з символьними обчисленнями.
- Pandas реалізує різні структури даних і аналіз.

Бібліотека scikit-learn складається з 35 модулів, які можна поділити на модулі кластеризації, модулі оцінки моделі і кількісного визначення якості прогнозів, модулі роботи з наборами даних (попередня обробка, нормалізація), модулі роботи з ознаками (витяг і виявлення найбільш значущих), модулі, які реалізують різні алгоритми рішення задач класифікації і регресії. Кожен модуль складається з класів і функцій і вирішує такі завдання, як:

- Кластеризація (Clustering) - групування нерозмічених даних.
- Перехресна перевірка (Cross Validation) - оцінка ефективності роботи моделі на незалежних даних.
- Набори даних - для зберігання тестових наборів даних і створення наборів даних з певними властивостями для дослідження поведінкових властивостей моделі.
- Зменшення розмірності - набір алгоритмів для зменшення кількості атрибутів для вибору ознак, наприклад, метод аналізу основних компонентів.
- Методи ансамблю - сукупність методів для об'єднання прогнозів декількох моделей.
- Вилучення функцій - процес визначення атрибутів у даних.
- Вибір функцій - набір алгоритмів для визначення відповідних атрибутів, на яких можна побудувати модель.
- Оптимізація налаштувань параметрів (налаштування параметрів) - методи для отримання найбільш ефективних результатів від моделі.
- Множинне навчання (Manifold Learning) - підхід нелінійного зменшення даних.

Окремо варто виділити методи навчання з вчителем (Supervised Models). Цей набір методів включає:

- узагальнені лінійні моделі (узагальнені лінійні моделі);
- методи дискримінаційного аналізу;
- наївний байєсівський класифікатор (Naive Bayes);
- нейронні мережі (нейронні мережі);
- метод опорних векторів (Support Vector Machines);
- дерева прийняття рішень (Decision Trees);

3.3 Бібліотека pandas

Pandas - це високорівнева Python бібліотека для аналізу даних, побудована поверх більш низькорівневої бібліотеки NumPy (написана на C), що є великим плюсом в продуктивності. В екосистемі Python, pandas є бібліотекою для обробки і аналізу даних, що найшвидше розвивається.

Бібліотека pandas надає дві структури: Series і DataFrame для швидкої і зручної роботи з даними (насправді їх три, є ще одна структура - Panel, але в даний момент вона перебуває в статусі deprecated і в майбутньому буде виключена зі складу бібліотеки pandas). Series - це маркована одномірна структура даних, її можна уявити, як таблицю з одним рядком. З Series можна працювати як зі звичайним масивом (звертатися за номером індексу), і як з асоційованим масивом, коли можна використовувати ключ для доступу до елементів даних. DataFrame - це двовимірна маркована структура. Ідейно вона дуже схожа на звичайну таблицю, що виражається в способі її створення і у роботі з її елементами. Panel - про який було сказано, що буде незабаром виключений з pandas, являє собою тривимірну структуру даних. В рамках цієї частини ми зупинимося на питаннях створення і отримання доступу до елементів даних структур Series і DataFrame.

Створити структуру Series можна на базі різних типів даних:

- словники Python;
- списки Python;
- масиви з numpy: ndarray;
- скалярні величини.

Найпростіший спосіб створити Series - це передати в якості єдиного параметра в конструктор класу список Python. Для доступу до елементів Series, у даному випадку, можна використовувати тільки позитивні цілі числа - лівий стовпець чисел, що починається з нуля - це як раз і є індекси елементів структури, які представлені в правій колонці.

Якщо Series являє собою одновимірну структуру, яку для себе можна уявити як таблицю з одним рядком, то DataFrame - це вже двовимірна структура - повноцінна таблиця з безліччю рядків і стовпців. Структуру DataFrame можна створити на базі:

- словника (dict) в якості елементів якого повинні виступати: одномірні ndarray, списки, інші словники, структури Series;
- двовимірні ndarray;
- структури Series;
- структуровані ndarray;
- інші DataFrame.

3.4 Платформа Anaconda

Anaconda - це платформа для наукових досліджень, заснована на мові програмування Python. Основна мета цього пакету - надати організаціям можливість успішно захищати, інтерпретувати, масштабувати і зберігати дані, які мають вирішальне значення для повсякденної роботи. За оцінками, понад 4,5 мільйонів користувачів вже завантажили цей пакет. Anaconda - це програмний пакет корпоративного рівня, який надає безліч інноваційних можливостей для кінцевого користувача. Кілька основних переваг включають

розширене міжвідомче співробітництво, здатність відтворювати дані, чудову масштабованість і кілька рівнів безпеки. Ще одна важлива особливість цього пакета полягає в тому, що він дозволяє організаціям краще управляти і інтерпретувати великі дані, що є ключовим фактором успіху в сучасному бізнес-середовищі. Він використовує кілька джерел даних, щоб гарантувати надмірність. До них відносяться (але не обмежуються ними) хмарні сховища, SQL, NoSQL.

Анаконда є модульною за своєю природою, тому її можна налаштувати в залежності від потреб відповідної організації. Так як це сприяє співробітництву в режимі реального часу, рівень власної ефективності також буде покращено. Користувачі можуть отримувати технічну підтримку в режимі реального часу під час розгортання відкритого коду і, оскільки Anaconda повністю сумісна з мовою Python, загальна крива навчання значно скоротиться.

3.5 Вибір мови програмування

Важко виділити якийсь конкретний інструмент, так як вся суть застосування технологій машинного навчання передбачає індивідуальний підхід до кожного завдання. При цьому для реалізації нашої системи, було проаналізовано декілька мов програмування та бібліотек за такими критеріями як статистичний аналіз, можливість первинного аналізу (побудова графіків та діаграм), математичне моделювання. Нами було виділено три основних претенденти це мови програмування Java, Python та C++ оскільки вони мають всі названі вище критерії. Для вибору найбільш підходящої мови програмування була побудована порівняльна таблиця.

Таблиця 1.1 – Таблиця порівняння мов програмування

	C++	Java	Python
Швидкість обробки	4	3	1
Читабельність	3	3	2
Простота	1	2	5
GUI (графічний інтерфейс користувача)	2	3	4
Графіка (2D)	3	3	5
Графіка (3D)	4	3	1
Кросплатформеність	3	5	5
Підсумок	20	22	23

Як бачимо по набраних балах лідирує мова програмування Python. У результаті, зважаючи на дані таблиці 1.1 та дивлячись на універсальність, широку підтримку, кросплатформеність, простоту і зрозумілість синтаксису, була обрана мова програмування Python, та бібліотеки для машинного навчання scikit-learn і для аналізу даних pandas відповідно. Для зручності роботи використовувалось середовище Anaconda, яке містить у собі названі вище пакети, та вбудований інтерпретатор Python.

Висновок до розділу 3

В даному розділі дипломного проекту були розглянуті інструменти для аналізу даних та побудови моделі прогнозування за допомогою алгоритмів машинного навчання. Був проведений огляд найбільш поширених інструментів для роботи з аналізом великих даних серед яких можна навести: мову програмування Python, бібліотеку для реалізації ряду алгоритмів машинного навчання scikit-learn, бібліотеку для обробки і аналізу даних pandas, платформу для наукових досліджень Anaconda.

4. ПРАКТИЧНА РЕАЛІЗАЦІЯ

Дослідження за допомогою методів машинного навчання ґрунтуються на даних, тому для отримання найбільш точних рішень, необхідно використовувати достовірні джерела інформації. Також важливу роль відіграє формування коректної структури даних для подальшої обробки за допомогою методів машинного навчання.

4.1 Структурна, функціональні схеми, UML діаграма розробленої системи та блок-схема алгоритму машинного навчання

Рис. 4.1. Функціональна схема системи

На рисунку 4.1 зображена функціональна схема системи моніторингу динаміки ринку, яка демонструє взаємодію компонетів нашого програмного забезпечення, показує інформаційні потоки та вказує на файли і пристрої, що використовуються. На схемі ми можемо спостерігати фактичний розподіл

нашої системи на дві підсистеми, тобто підсистема обробки та аналізу даних та підсистема прогнозування.

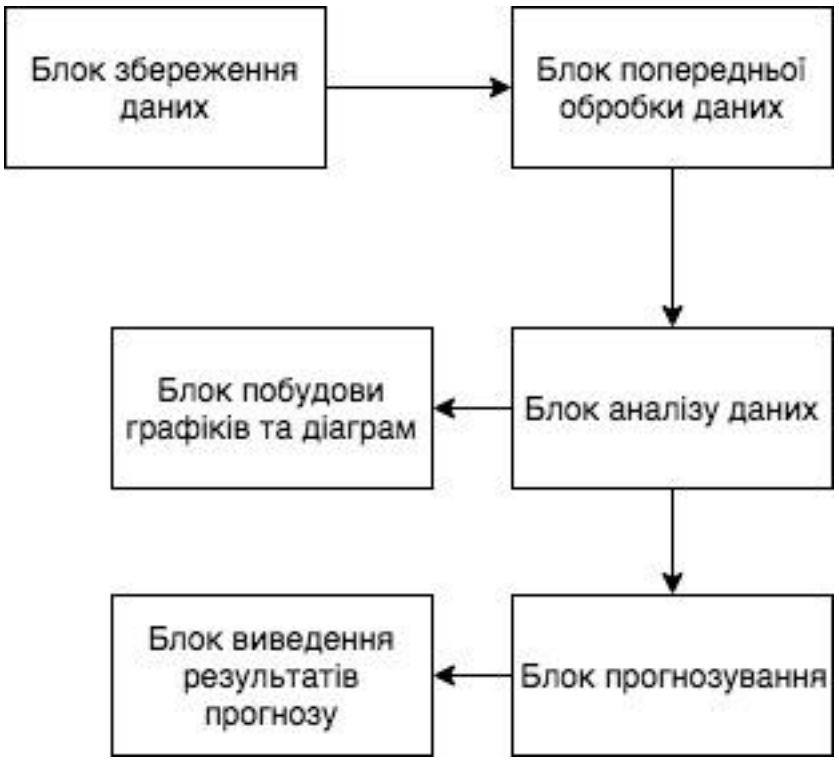

Рис. 4.2. Структурна схема системи

Ми можемо спостерігати складові нашої системи на рисунку 4.2, тобто основні блоки та головні зв’язки між ними. Із даної схеми, можна зрозуміти як працює наша система в основних режимах роботи, тобто спочатку відбувається попередня обробка даних, потім аналіз і побудова діаграм, і у фіналі прогнозування і вивід результатів.

Рис. 4.3. UML діаграма станів

На рисунку 4.3 зображена UML діаграма станів, яка визначає зміну нашої системи у часі. Коло позначає початковий стан системи, а коло з окружністю усередині – кінцевий стан.

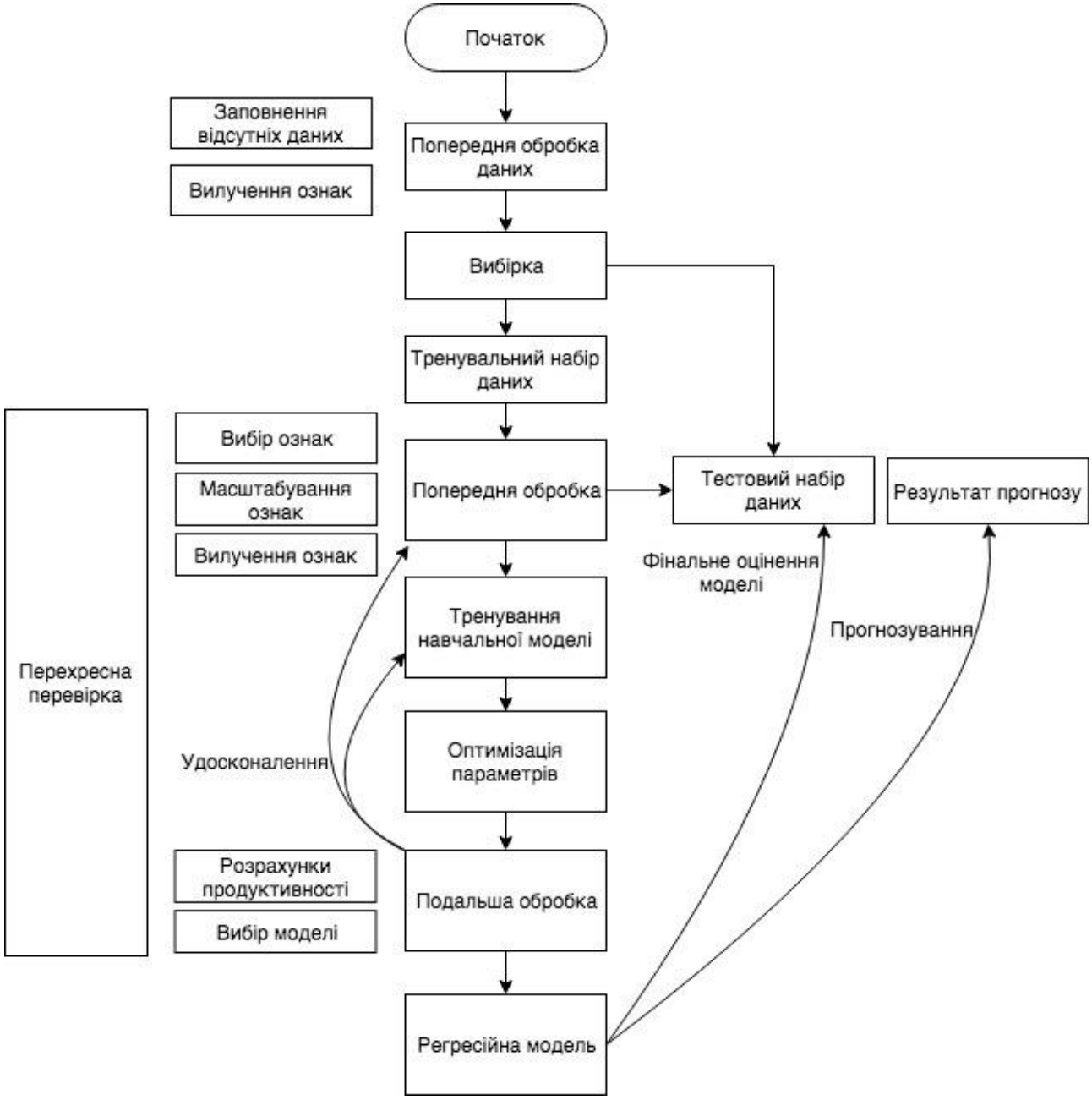

Рис. 4.4. Блок-схема алгоритму машинного навчання (модель прогнозування цін)

Блок-схема алгоритму машинного навчання, що зображена на рисунку 4.4 детально, крок за кроком, демонструє, підготовку даних, побудову та навчання нашої моделі прогнозування цін.

4.2 Формування даних для дослідження

Для використання методів машинного навчання для моніторингу динаміки автомобільного ринку та прогнозування вартості автомобілів було сформовано вибірку проданих автомобілів з 1990 – 2017 роки. Дані вибірки були отримані із ресурсу kaggle.com, який представляє собою платформу де зібрані набори даних із різних галузей. На рисунку 4.5 представлений інтерфейс даної платформи.

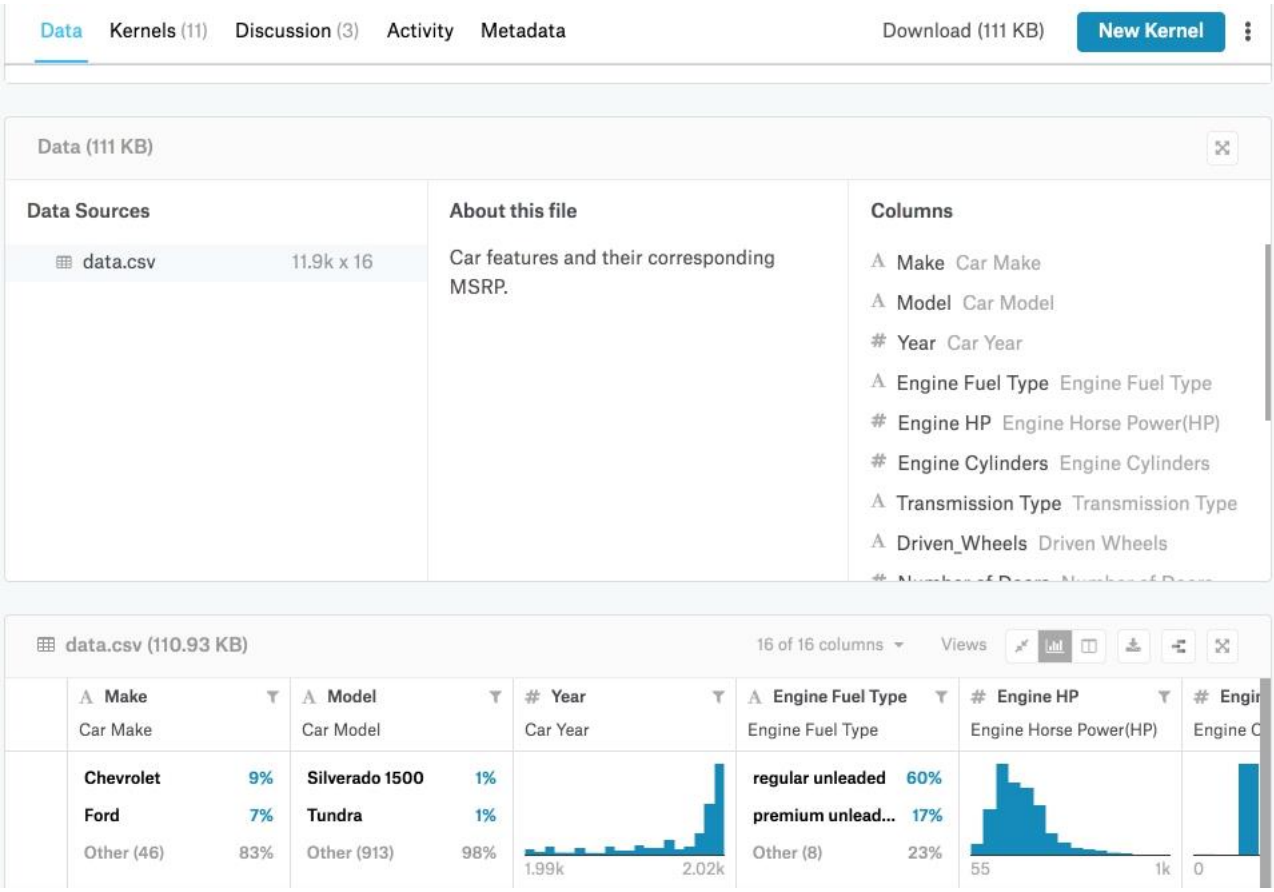

Рис. 4.5. Інтерфейс платформи kaggle.com

Вибірка даних містить більше 11000 прикладів проданих автомобілів, і деякі їхні характеристики. Набір даних був сформований у форматі CSV (Comma-Separated Values), для кожного автомобіля були наявні наступні параметри:

- марка;
- модель;

- рік випуску;
- тип палива;
- потужність двигуна;
- кількість циліндрів двигуна;
- тип трансмісії;
- тип приводу;
- кількість дверей;
- категорія ринку;
- розмір транспортного засобу;
- стиль транспортного засобу;
- розхід пального на шосе;
- розхід пального у місті;
- популярність;
- ціна.

Приклад фрагмента вихідного файлу вибірки на малюнку 4.6.

	Make	Model	Year	Engine Fuel Type	Engine HP	Engine Cylinders	Transmission Type	Driven_Wheels	Number of Doors	Market Category	Vehicle Size	Vehicle Style	highway MPG	city mpg
0	BMW	1 Series M	2011	premium unleaded (required)	335.0	6.0	MANUAL	rear wheel drive	2.0	Factory Tuner,Luxury,High-Performance	Compact	Coupe	26	19
1	BMW	1 Series	2011	premium unleaded (required)	300.0	6.0	MANUAL	rear wheel drive	2.0	Luxury,Performance	Compact	Convertible	28	19
2	BMW	1 Series	2011	premium unleaded (required)	300.0	6.0	MANUAL	rear wheel drive	2.0	Luxury,High-Performance	Compact	Coupe	28	20
3	BMW	1 Series	2011	premium unleaded (required)	230.0	6.0	MANUAL	rear wheel drive	2.0	Luxury,Performance	Compact	Coupe	28	18
4	BMW	1 Series	2011	premium unleaded (required)	230.0	6.0	MANUAL	rear wheel drive	2.0	Luxury	Compact	Convertible	28	18

Рис. 4.6. Фрагмент вибірки даних автомобілів

4.3 Передобробка даних

Підготовка даних - важливий етап в машинному навчанні. Використання якісно підготовлених даних підвищує точність прогнозів навіть при використанні простих моделей.

На першому етапі підготовки необхідно перетворити дані, специфічні для предметної області, в зрозумілі для моделі. Наступний етап підготовки даних - адаптація набору даних під вимоги алгоритму. Для метричних методів необхідна попередня нормалізація кількісних ознак для коректної оцінки їх впливу на цільову змінну. При використанні лінійних моделей необхідна попередня стандартизація.

Було проведено дослідження усіх наявних параметрів та знайдено усі пусті значення за допомогою інструментів бібліотеки pandas. На рисунку 4.7 наведено список усіх наявних параметрів.

```
RangeIndex: 11914 entries, 0 to 11913
Data columns (total 16 columns):
Make                 11914 non-null object
Model                11914 non-null object
Year                 11914 non-null int64
Engine Fuel Type     11911 non-null object
Engine HP            11845 non-null float64
Engine Cylinders     11884 non-null float64
Transmission Type    11914 non-null object
Driven_Wheels        11914 non-null object
Number of Doors      11908 non-null float64
Market Category      8172 non-null object
Vehicle Size         11914 non-null object
Vehicle Style        11914 non-null object
highway MPG          11914 non-null int64
city mpg             11914 non-null int64
Popularity           11914 non-null int64
Price                11914 non-null int64
dtypes: float64(3), int64(5), object(8)
memory usage: 1.5+ MB
```

Рис. 4.7. Список параметрів

На рисунку 4.8 наведено розподіл відсутніх даних, тобто у правій колонці у нас находяться параметри, а у лівій відповідно кількість пустих значень.

```
Make                    0
Model                   0
Year                    0
Engine Fuel Type        3
Engine HP              69
Engine Cylinders       30
Transmission Type       0
Driven_Wheels           0
Number of Doors         6
Market Category      3742
Vehicle Size            0
Vehicle Style           0
highway MPG             0
city mpg                0
Popularity              0
Price                   0
dtype: int64
```

Рис. 4.8. Розподіл відсутніх значень

Найбільша кількість відсутніх даних припала на категорію ринку “Market category” 3742 екземпляри, що становить майже 1/3 набору даних. Оскільки 1/3 є досить великою кількістю відсутніх значень, було прийнято рішення не присвоювати середні значенню всього набору даних.

Замість цього було створено функцію, яка обчислює ринкову категорію яка найчастіше зустрічається для кожної моделі транспортного засобу, де є відсутнє значення, і замінює ці відсутні дані.

Також багато моделей взагалі не мали жодної ринкової категорії. Було вирішено, залишити такі рядки, оскільки вони утворюють досить великий набір даних, а функція "ринкова категорія" - змінна, яка не буде безпосередньо використана в нашій регресійній моделі.

Було представлено три транспортні засоби з відсутнім параметром тип палива “Engine Fuel Type”. Відсутні значення відповідають трьом Suzuki Verona. Дивлячись тільки на цей вид транспортних засобів, можна визначити, що всі вони відповідають автомобілям, що працюють на звичайному неетильованому паливі. Ми замінили відсутні значення на неетильоване паливо для даних екземплярів.

Також, було виявлено велику кількість автомобілів із відсутнім значенням параметру кількість циліндрів двигуна “Engine Cylinders”, або зі значенням 0. Велика кількість нульових значень, пов'язана з електричними автомобілями. Дана категорія автомобілі також не має інших числових змінних, які потрібні нам для того, щоб побудувати прогнозну модель (потужність двигуна, кількість дверей). Тому було прийняте рішення їх видалити. Відсутність значень для колонки “Engine Cylinders” стосується Mazda RX7 та RX8 (у даних моделях використовується роторний двигун, який не має циліндрів). Ми замінили відсутні значення на середні для даних авто.

Відсутнє значення кількості дверей було представлене тільки у Ferrari FF, оскільки даний «спорт кар» має двоє дверей, було замінено відсутнє значення на відповідне.

Було виявлено велику кількість відсутніх значень для параметра потужність двигуна “Engine horsepower”. Як і у випадку параметра ринкової категорії, нами було використано функцію для заміни відсутніх значень середнім значенням потужності для відповідної моделі.

На рисунку 4.9 представлено набір даних без відсутніх значень, за винятком категорії ринку, яку ми залишили без змін.

```
Make                    0
Model                   0
Year                    0
Engine Fuel Type        0
Engine HP               0
Engine Cylinders        0
Transmission Type       0
Driven_Wheels           0
Number of Doors         0
Market Category      2438
Vehicle Size            0
Vehicle Style           0
highway MPG             0
city mpg                0
Popularity              0
Price                   0
dtype: int64
```

Рис. 4.9. Розподіл відсутніх значень після проведеної обробки даних

Також, був доданий параметр “Age”, щоб оцінити, наскільки старий автомобіль, що є дуже важливою функцією для тих, хто хоче придбати автомобіль. Ми використовуємо 2017 рік, як поточний, оскільки набір даних був сформований у цьому.

У кінці передобробки даних ми перетворили споживання палива на шосе та у місті з миль на галон у літри на 100 кілометрів. Для цього ми поділили 235 на значення колонки витрат пального у милях на галон.

У результаті попередньої обробки, дані були приведені до вигляду, зручного для подальшої роботи з методами машинного навчання. Тепер у нас є чистий і повний набір даних, над якими ми можемо працювати. Фрагмент статистичних даних, представлений на рисунку 4.10.

Year	Engine Fuel Type	Engine HP	Engine Cylinders	Transmission Type	Driven_Wheels	Number of Doors	Market Category	Vehicle Size	Vehicle Style	Highway L/100km	City L/100km	Popularity	Price	Age
2011	premium unleaded (required)	335	6.0	MANUAL	rear wheel drive	2.0	Factory Tuner,Luxury,High-Performance	Compact	Coupe	9.04	12.37	3916	46135	6
2011	premium unleaded (required)	300	6.0	MANUAL	rear wheel drive	2.0	Luxury,Performance	Compact	Convertible	8.39	12.37	3916	40650	6
2011	premium unleaded (required)	300	6.0	MANUAL	rear wheel drive	2.0	Luxury,High-Performance	Compact	Coupe	8.39	11.75	3916	36350	6
2011	premium unleaded (required)	230	6.0	MANUAL	rear wheel drive	2.0	Luxury,Performance	Compact	Coupe	8.39	13.06	3916	29450	6
2011	premium unleaded (required)	230	6.0	MANUAL	rear wheel drive	2.0	Luxury	Compact	Convertible	8.39	13.06	3916	34500	6

Рис. 4.10. Фрагмент вибірки даних після обробки

4.4 Дослідження даних

Основні статистичні характеристики числових даних (марка, модель, рік випуску, тип палива, потужність двигуна, кількість циліндрів двигуна, тип трансмісії, тип приводу, кількість дверей, категорія ринку, розмір транспортного засобу, стиль транспортного засобу, розхід пального на шосе, розхід пального у місті, популярність, ціна, вік) представлені на малюнку 4.6. Аналізуючи ці дані, можемо зробити висновок, що ми маємо повний набір даних (кількість записів є однаковою по кожній колонці, що говорить про відсутність пропусків у даних, та їх повноту).

Щоб провести аналіз даних, нами було побудовано декілька залежностей, на основі яких було зроблені певні висновки. Першою було побудовано

залежність ціни транспортного засобу від потужності двигуна. Дана залежність зображена на рисунку 4.11.

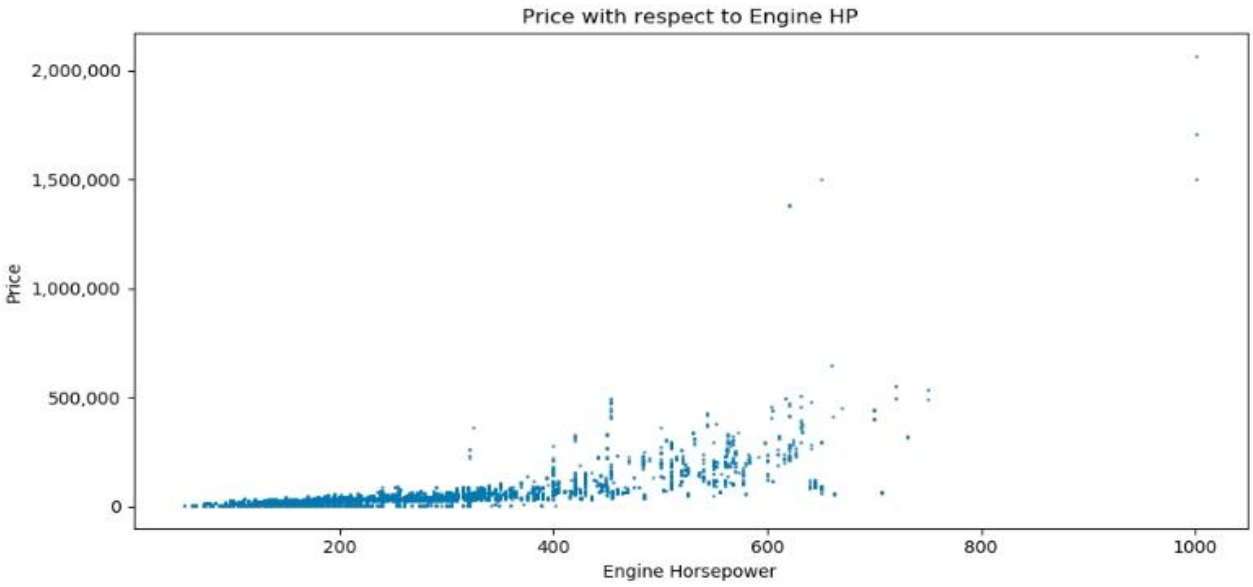

Рис. 4.11. Залежність ціни від потужності двигуна

Ми можемо чітко спостерігати експоненціальний зв'язок між цими змінними. У наступних розділах ми поговоримо про це, і побачимо, яка залежність (логарифмічна, квадратна, експоненційна) пропонує найкращу кореляцію.

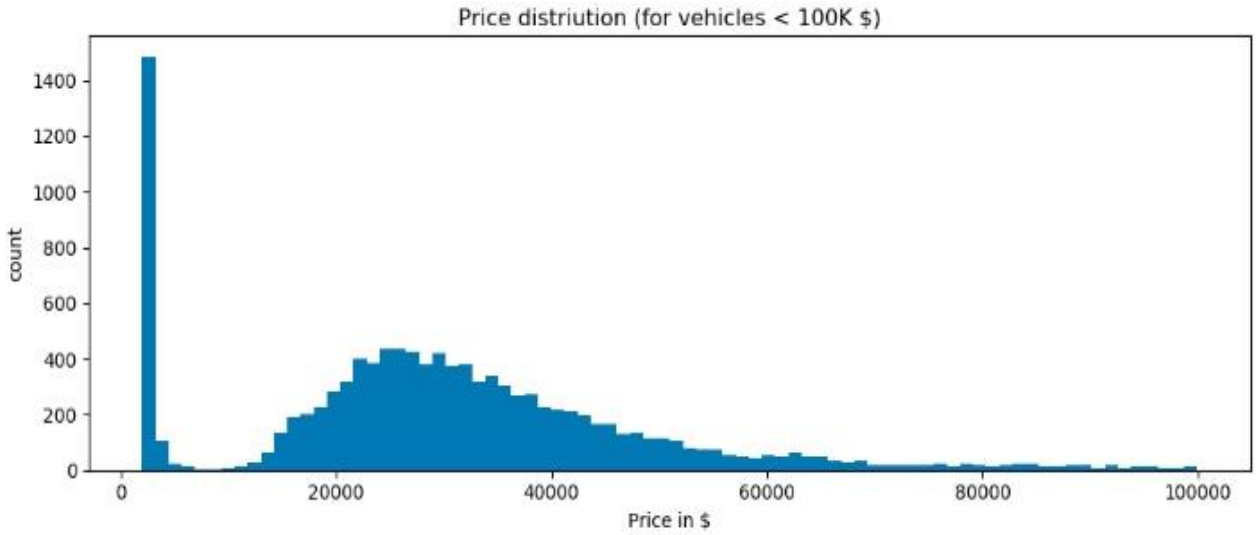

Рис. 4.12. Графік розподілу цін

Далі ми розбиваємо дані для цілей візуалізації. Ми виділяємо категорію до 100000$, оскільки вона охоплює 95% усіх екземплярів автомобілів, та будуємо графік розподілу цін, який зображено на рисинку 4.12.

Проаналізувавши графік розподілу цін, можна зробити висновок, що ціни на автомобілі не мають нормального розподілу, окрім асиметричності правої частини графіку цін, ми можемо відзначити цілу групу автомобілів, що конкурують на низькому ціновому сегменті ринку.

Наступним кроком, була побудова залежності ціни від кількості циліндрів двигуна для цінового сегменту до 100 000$ та більше 100 000$. Були отримані результати зображені на рисунку 4.13.

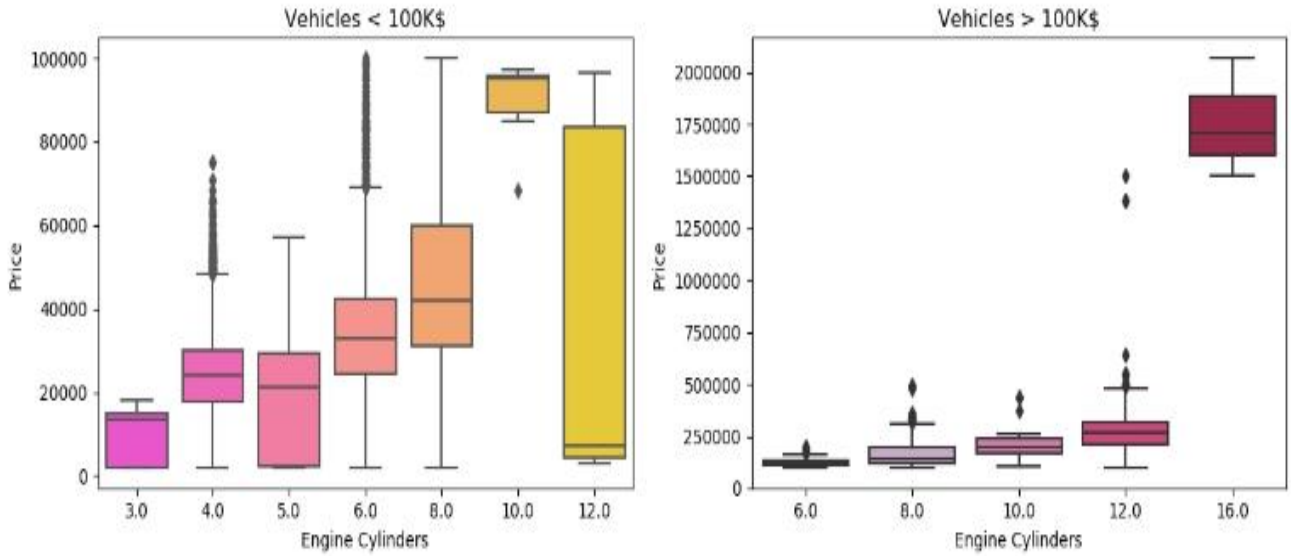

Рис. 4.13. Графік діапазону цін в залежності від розміру двигуна

Як і очікувалося, ціна рухається вгору разом з кількістю циліндрів двигуна. Це пояснюється тим, що лінійні залежності можуть бути не кращим поясненням мінливості ціни, заснованої на кількості циліндрів двигуна.

Проте, діапазон значень, 12-циліндрових автомобілів (для автомобілів вартістю менше 100 000 $), несподівано розширився.

Аналіз цього типу транспортних засобів показує, що низькі значення вартості (3000 - 7500 $) відповідають старим BMW і Mercedes 90-х років, але кілька Volkswagen Phaeton продаються по 80K - 100K, звідси і мінливість цін.

4.5 Кореляційний аналіз

Матриця кореляції числових ознак - вид матриці даних, що включає коефіцієнти кореляції для всіх пар аналізованих змінних. Матриця кореляції являє собою основу для факторного аналізу, канонічної кореляції та інших статистичних технік, що відтворюють структуру залежності між змінними.
Візуальне відображення матриці кореляції ознак, що використовуються для прогнозу цін на автомобілі наведено на рисунку 4.14.

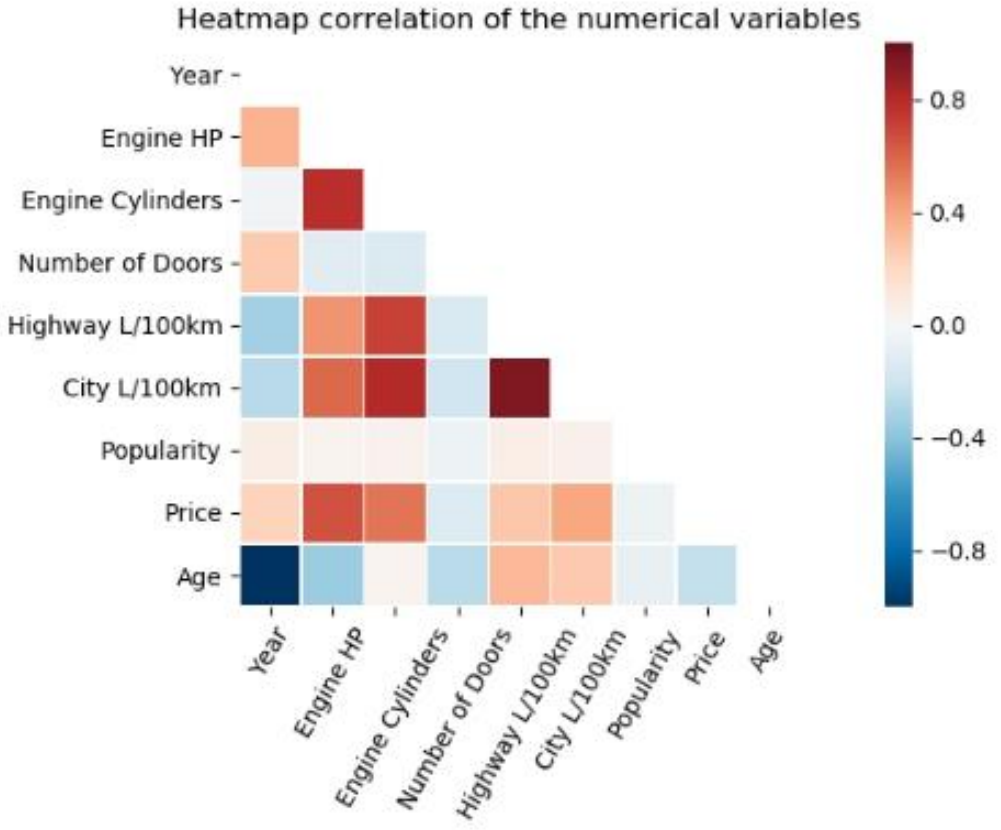

Рис. 4.14. Матриця кореляції вихідних ознак

Як бачимо, існує дуже сильна кореляція між змінними “City L / 100km” та “Highway L / 100km”, що становить 0.94 (Рисунок 4.11). Це має сенс,

оскільки ці обидві змінні вимірюють споживання палива автомобіля, яке, ймовірно, буде пропорційним в обох ситуаціях.

Можна також відзначити ідеальну негативну кореляцію (-1) між віком і роком, що є абсолютно нормальним, оскільки параметр «вік» випливає з року.

	Year	Engine HP	Engine Cylinders	Number of Doors	Highway L/100km	City L/100km	Popularity	Price	Age
Year	1.000000	0.353345	-0.032725	0.262659	-0.328514	-0.270576	0.072678	0.227569	-1.000000
Engine HP	0.353345	1.000000	0.781364	-0.102205	0.454313	0.587292	0.035649	0.662392	-0.353345
Engine Cylinders	-0.032725	0.781364	1.000000	-0.136513	0.716284	0.809878	0.045831	0.547048	0.032725
Number of Doors	0.262659	-0.102205	-0.136513	1.000000	-0.148169	-0.180514	-0.049396	-0.127644	-0.262659
Highway L/100km	-0.328514	0.454313	0.716284	-0.148169	1.000000	0.941064	0.069868	0.275555	0.328514
City L/100km	-0.270576	0.587292	0.809878	-0.180514	0.941064	1.000000	0.051949	0.395896	0.270576
Popularity	0.072678	0.035649	0.045831	-0.049396	0.069868	0.051949	1.000000	-0.048419	-0.072678
Price	0.227569	0.662392	0.547048	-0.127644	0.275555	0.395896	-0.048419	1.000000	-0.227569
Age	-1.000000	-0.353345	0.032725	-0.262659	0.328514	0.270576	-0.072678	-0.227569	1.000000

Рис. 4.15. Таблиця кореляції

Провівши кореляційний аналіз, ми можемо навести 5 ключових аргументів для використання лінійної регресії, при побудові моделі для прогнозування цін на автомобілі:

- лінійні відносини;
- багатовимірна нормальність;
- немає або мало мультиколінеарності;
- немає автокореляції;
- гомоседастичність (однорідна варіативність значень спостережень, що виражається у відносній стабільності);

З таблиці кореляції ми можемо виділити 8 змінних, які мають залежності з нашою цільовою змінною “Price”.

Таблиця 1.2 – Залежні змінні з цільовою змінною

	Year	Engine HP	Engine Cylinders	Number of Doors	Highway L/100km	City L/100km	Popular	Age
Price	0.227	0.662	0.547	-0.127	0.275	0.395	-0.048	-0.227

Проаналізувавши Таблицю 1.2 ми виділили змінні, які мають сильнішу залежність із цільовою змінною, у своїй області характеристик, і котрі ми будемо використовувати у нашій моделі. Відповідно для побудови нашої моделі ми вибрали змінну в якості параметра, що характеризує двигун “Engine HP”, оскільки дана змінна має сильнішу кореляцію ніж “Engine Cylinders” 0.662 проти 0.547. В якості змінної, що характеризую витрати пального, нами була вибрана “City L/100km”, а в якості вікової характеристики ми обрали змінну “Age”.

Таблица 1.3 – Кореляція змінних, що використовуються для нашої моделі

	Engine HP	City L/100km	Age
Price	0.662	0.395	-0.227

Мультиколінеарність - це співвідношення між змінними. Деякі наші параметри, суперечать цьому припущенню. Крім того, таблиця кореляції вказує на те, що споживання палива в місті є кращим показником для ціни автомобіля, ніж споживання палива на шосе. Нарешті, параметр рік є надлишковим з віком автомобіля. Тому було прийнято рішення видалити змінні “Highway L/100km” та “Year”.

Раніше ми говорили, що розподілення цін було сильно спотворено, тому ми змушені були розділити дані, щоб мати значущі ділянки. Додавши до цього те, що ціна часто має експоненціальне відношення до інших змінних (особливо

потужність двигуна, кількість циліндрів двигуна, витрати пально у місті на 100 км), ми будемо логарифмувати змінну ціни, а пізніше обчислимо коефіцієнти кореляції між логарифмом ціни та різними функціями потужності двигуна.

І нарешті, що до кількісті дверей, то ця змінна більш доречна, якщо вона використовується як категоріальна змінна, тому ми не будемо використовувати її в нашій регресійній моделі.

	Price	Log_price
Engine HP	0.662392	0.680113
Sq_engine_hp	0.743719	0.623111
Sqrt_engine_hp	0.609975	0.692852
Log_engine_hp	0.552919	0.693258

Рис. 4.16. Таблиця кореляції ціни та потужності

Найбільш сильною кореляцією із змінною Log_price є використання Log_engine_hp (0.693). Ми видалили інші функції потужності.

	Price	Log_price
Age	-0.227569	-0.768184
Sq_age	-0.246195	-0.820073
Sqrt_age	-0.185399	-0.663697
Log_age	-0.163754	-0.603754

Рис. 4.17. Таблиця кореляції ціни та віку

Квадрат змінної віку значно більше корелює з Log_price, ніж коли змінні не були трансформовані (-0,82 проти -0,22). Тому ми зберігаємо Sq_age.

	Price	Log_price
City L/100km	0.395896	0.151824
Sq_city	0.458524	0.170509
Sqrt_city	0.363130	0.141367
Log_city	0.334592	0.131878

Рис. 4.18. Таблиця кореляції ціни та віку

Виберемо стовпець Sq_city, який дозволяє максимально збільшити коефіцієнт кореляції з Log_price.

У підсумку отримаємо числові змінні, які ми будемо використовувати для нашої моделі: Log_price, Log_engine_hp, Sq_age, Sq_city.

	Log_price	Log_engine_hp	Sq_age	Sq_city
0	10.739349	5.817111	36	153.0169
1	10.612779	5.707110	36	153.0169
2	10.500977	5.707110	36	138.0625
3	10.290483	5.442418	36	170.5636
4	10.448744	5.442418	36	170.5636

Рис. 4.19. Фрагмент вибірки підготовлених даних

4.6 Застосування методів машинного навчання

У даній роботі для прогнозування цін на автомобілі використовувалися такі методи машинного навчання як лінійна та поліноміальна регресія.

Лінійна регресія - метод відновлення лінійної залежності. Регресійний аналіз має на увазі статистичне дослідження впливу однієї або декількох незалежних змінних (предикторів, ознак) на залежну (цільову) змінну. Цільова

змінна y в такому випадку представляється у вигляді лінійної комбінації предикторів (x1, x2, .., xn).

$$y(w, x) = w_0 + w_1 x_1 + \cdots + w_p x_p,$$

де w_p – коефіцієнти регресії.

Лінійна регресія налаштовує лінійну модель за допомогою коефіцієнтів регресії за принципом мінімізації залишкової суми квадратів між відповідями набору даних і відповідями, передбаченими лінійним наближенням.

$$\min_w ||X_w - y||^2$$

Оцінка за допомогою методів найменших квадратів дуже чутлива до випадкових помилок, тому важливо, щоб вхідні ознаки були незалежними (мали слабку кореляцію).

Поліноміальна регресія у свою чергу означає наближення даних (xi, yi) поліномом k-го ступеня A (x) = a + b - x + c - x2 + d - x3 + ... + h - xk. При k = 1 поліном описують прямою лінією, при k = 2 - параболою, при k = 3 - кубічною параболою і т.д. Як правило, на практиці застосовують k < 5. Також, потрібно зважити на те, що для побудови регресії поліномом k-го ступеня необхідна наявність принаймні (k + l) точок даних.

Поліноміальна регресія корисна для опису характеристик, що мають кілька яскраво виражених екстремумів (максимумів і мінімумів). Вибір ступеня поліному визначається кількістю екстремумів досліджуваної характеристики. Так, поліном другого ступеня може добре описати процес, що має тільки один максимум чи мінімум; поліном третього ступеня – не більше двох екстремумів; поліном четвертого ступеня – не більше трьох екстремумів і т.д.

4.6.1 Проста лінійна регресія

Для побудови моделі прогнозування на основі алгоритму лінійної регресії, у першу чергу ми визначили змінну предиктор (Log_engine_hp) та цільову змінну (Log_price). Далі було розподілено набір даних на тренувальний і тестовий. Ми використовували 80% набору для навчання і 20% для тестування.

На рисунку 4.20 представлена залежність цільової змінної Y (вісь абсцис) від предиктора X (вісь ординат). На малюнку точками позначені значення вибірки, лінія відображає регресійну залежність y(x) = 1.8348x + 0.1439.

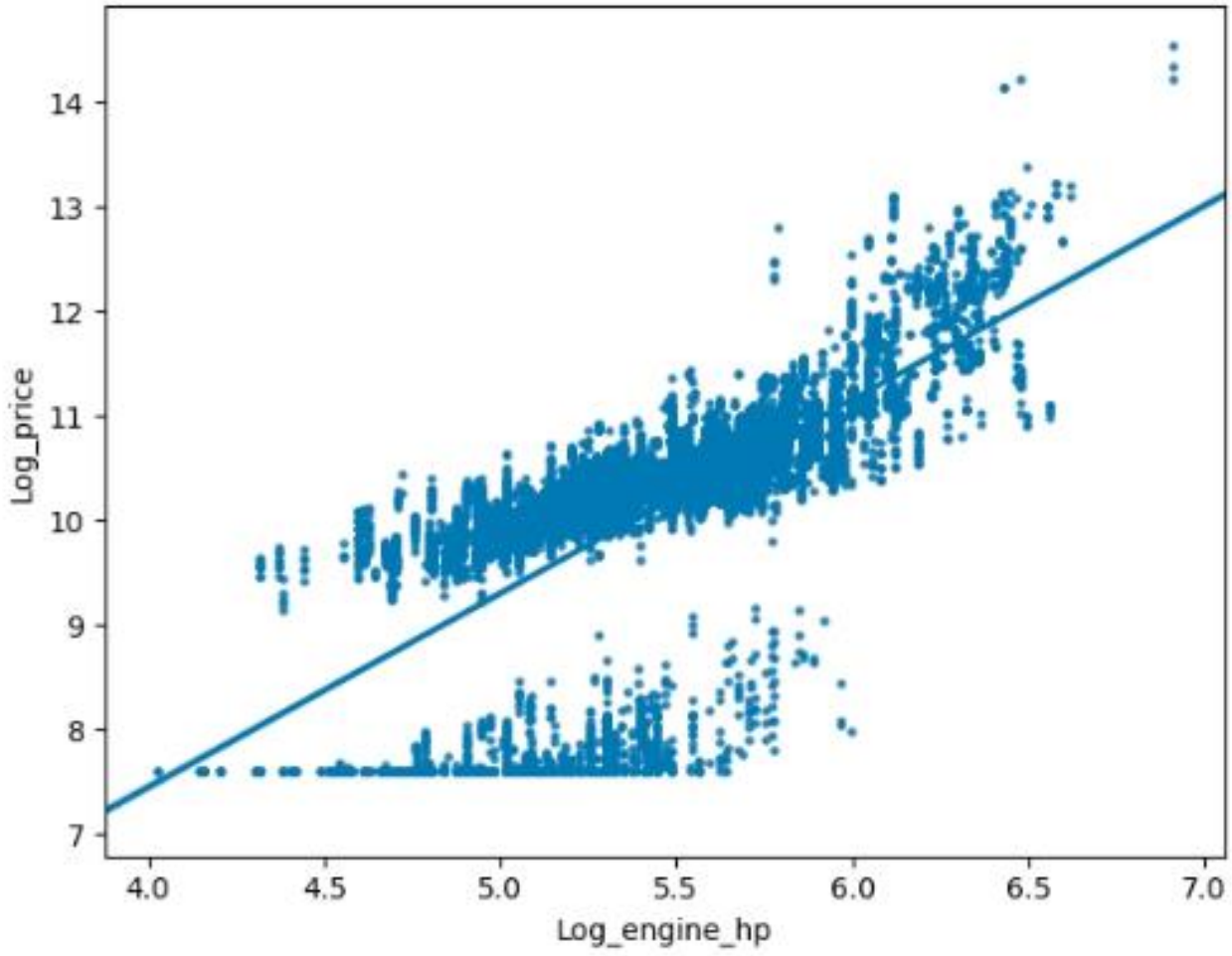

Рис. 4.20. Лінійна регресія

Таблиця 1.4 – Результати лінійної регресії

Вибірка	Точність передбачення, %
Навчальна	47,79
Тестова	48,89

Для прикладу використання нашої моделі, було зроблено прогноз для 1-го екземпляру із нашої вибірки проданих автомобілів. Отримане значення вартості становило 76980.0, в той час як актуальне значення становило 46135. Актуальні значення вибірки представлені на рисунку 4.17.

	Make	Model	Year	...	Popularity	Price	Age
0	BMW	1 Series M	2011	...	3916	46135	6
1	BMW	1 Series	2011	...	3916	40650	6
2	BMW	1 Series	2011	...	3916	36350	6
3	BMW	1 Series	2011	...	3916	29450	6
4	BMW	1 Series	2011	...	3916	34500	6

Рис. 4.17 Фрагмент актуальних значень вибірки

На рисунку 4.21 представлений графік розподілу фактичних і прогнозованих значень.

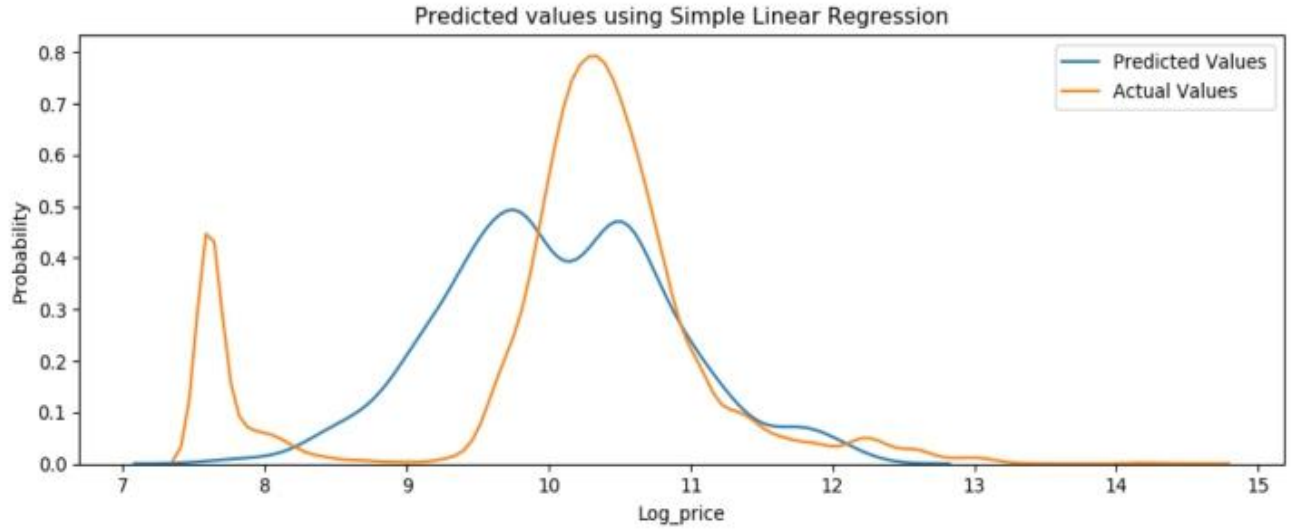

Рис. 4.21. Розподіл фактичних і прогнозованих значень

Графік розподілу демонструє, що ця проста лінійна модель має тенденцію недооцінювати частину транспортних засобів низького цінового діапазону (пік розбіжностей для автомобілів вартістю близько 3000 $).

Проста модель лінійної регресії також не працює для преміум-сегменту (автомобілі вище 270 000 $ (exp (12,5)).

4.6.2 Поліноміальна регресія

Для побудови моделі прогнозування на основі алгоритму поліноміальної регресії, ми визначили змінну предиктор (Engine HP) та цільову змінну (Price). Розподілили набір даних на тренувальний і тестовий. Ми використовували 70% набору для навчання і 30% для тестування.

На рисунку 4.22 представлена залежність цільової змінної Y (вісь абсцис) від предиктора X (вісь ординат). Лінія відображає поліноміальну регресійну залежність 3-го ступеня $y(x) = 0.003071\ x^3 - 2.101\ x^2 + 628.1\ x - 4.01e{+}04$.

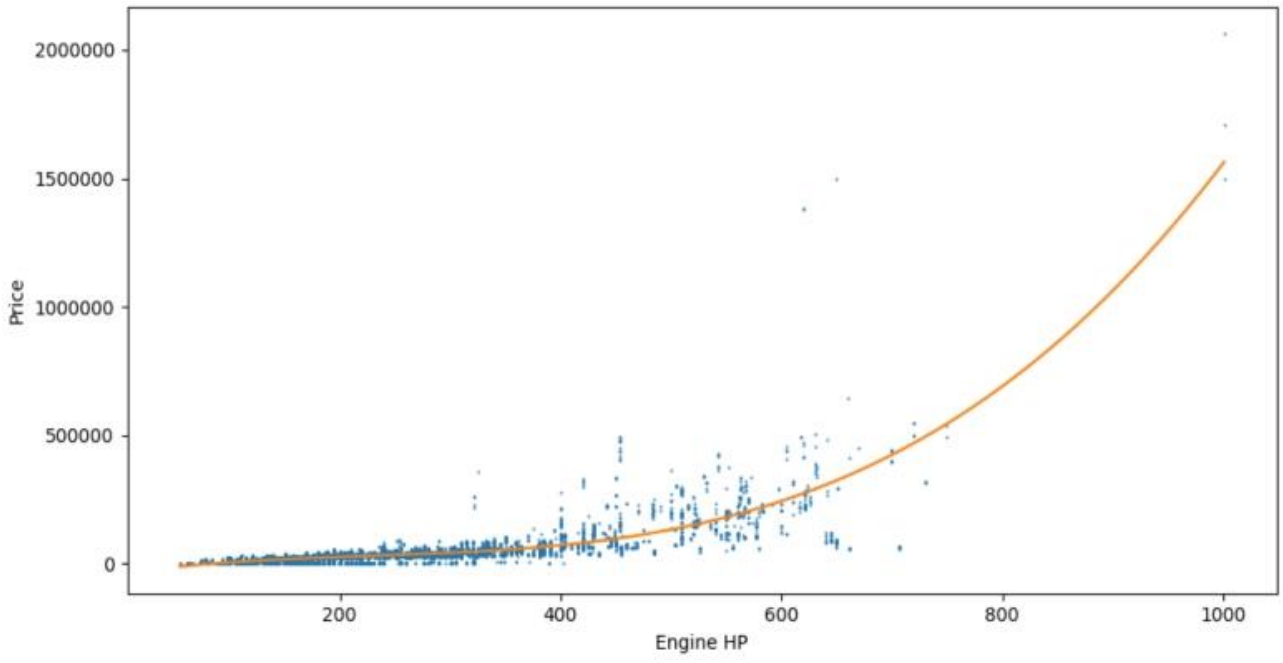

Рис. 4.22. Поліноміальна регресія

Таблиця 1.5 – Результати лінійної регресії

Вибірка	Точність передбачення, %
Навчальна	66,23
Тестова	54,15

Зробивши прогноз для 1-го екземпляру із нашої вибірки проданих автомобілів. Отримане значення вартості становило 80675, в той час як актуальне значення становило 46135. Незважаючи на це, модель поліноміальної регресії робить кращі прогнози ціни транспортного засобу, ніж модель простої лінійної регресії (SLR). Ми бачимо, що точність передбачення вища, ніж SLR (54% проти 48,89%), що означає вона може прогнозувати більш мінливі дані.

На рисунку 4.23 представлений графік розподілу фактичних і прогнозованих значень.

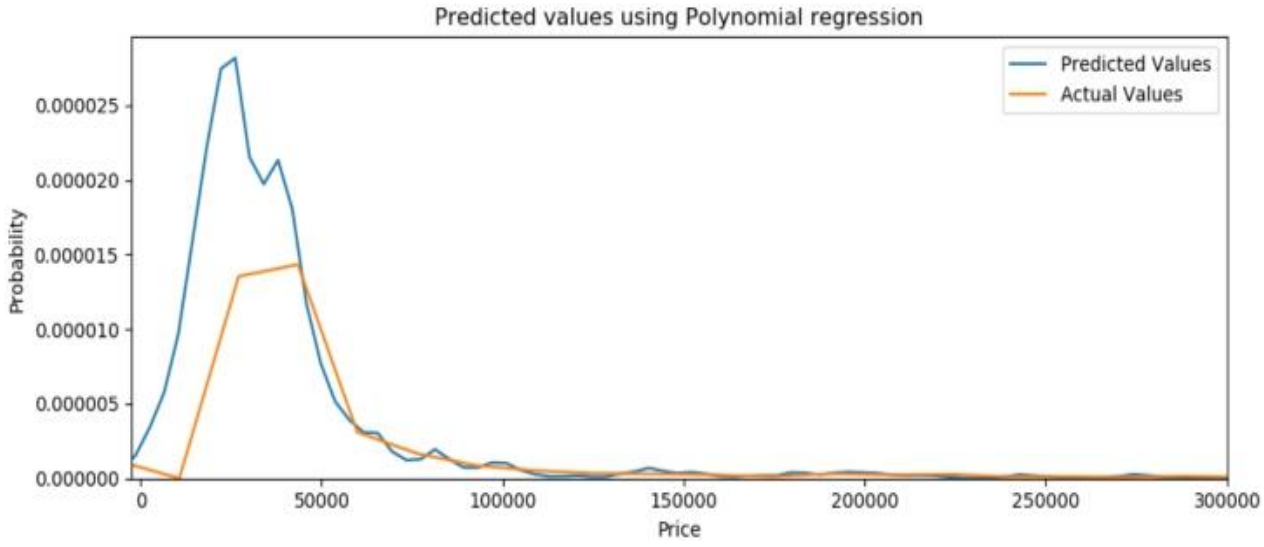

Рис. 4.23. Розподіл фактичних і прогнозованих значень

Проаналізувавши графік можна зробити висновок, що наша поліноміальна модель має тенденцію до надмірної оцінки частини автомобілів у ціновому діапазоні нижче 35 тисяч доларів США.

4.6.3 Множинна лінійна регресія

Для побудови моделі прогнозування на основі алгоритму множинної лінійної регресії, нам потрібно визначити декілька змінних предикторів. У нашому випадку це будуть “Log_engine_hp”, “Sq_age”, “Sq_city” та цільову змінну “Log_price”. Розподіл набору даних на тренувальний і тестовий, ми виконали у наступному вигляді 70% набору для навчання і 30% для тестування.

Таблиця 1.6 – Результати множинної лінійної регресії

Вибірка	Точність передбачення, %
Навчальна	83,57
Тестова	82,93

Функція множинної лінійної регресії: $y(x) = 0.8949\ x_1 - 0.0044\ x_2 + 0.0014\ x_3 - 5.44$.

Зробивши прогноз для 1-го екземпляру із нашої вибірки автомобілів, отримали значення 36002.0, в той час як актуальне значення 46135.

Використання 3-х змінних, допомагає робити кращі прогнозування. Дивлячись на точність передбачення нашої моделі ми можемо передбачити майже 83% варіацій цін, що є великим поліпшенням порівняно з простою лінійною регресією 48% або поліноміальною регресією з 1 ознакою 54%.

На рисунку 4.24 представлений графік розподілу фактичних і прогнозованих значень для множинної лінійної регресії.

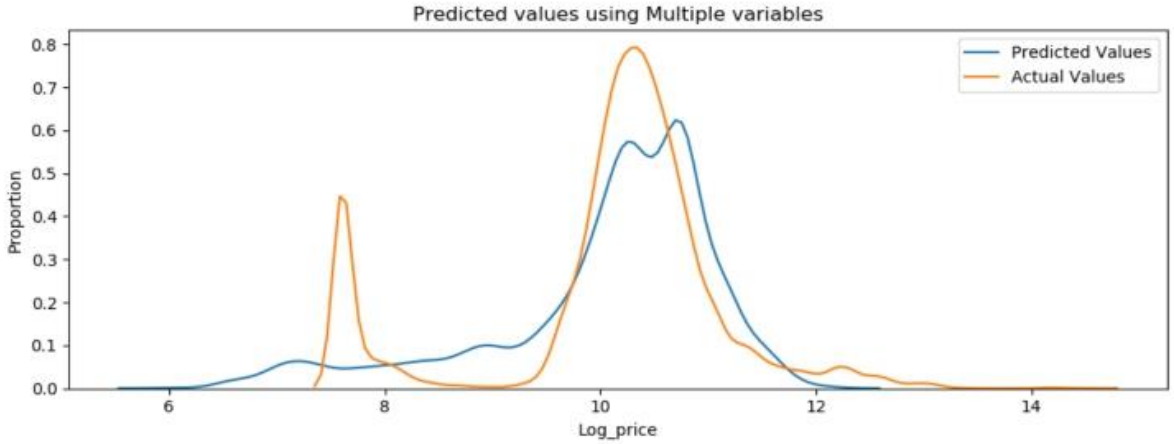

Рис. 4.24. Розподіл фактичних і прогнозованих значень

Проаналізувавши розподіл фактичних і прогнозованих значень помічаємо чітке поліпшення результатів прогнозування: ми маємо справедливу оцінку вартості автомобілів у ціновому діапазоні понад 22000 доларів (e ^ 10)). Однак наша модель все ще не точно передбачає ціну недорогих автомобілів (особливо автомобілів цінового діапазону близько 1800 $ (e ^ 7,5)).

4.6.4 Множинна поліноміальна регресія

Для побудови моделі прогнозування на основі алгоритму множинної поліноміальної регресії, ми визначили декілька змінних предикторів. У нашому випадку це будуть “Engine HP”, “Age”, “City L/100km” та цільову змінну “Price”. Розподіл набору даних на тренувальний і тестовий, ми виконали у наступному вигляді 70% набору для навчання і 30% для тестування.

Таблиця 1.7 – Результати поліноміальної множинної регресії

Вибірка	Точність передбачення, %
Навчальна	88,74
Тестова	84,79

Зробивши прогноз для 1-го екземпляру із нашої вибірки автомобілів, отримали значення 42239.0, в той час як актуальне значення 46135.

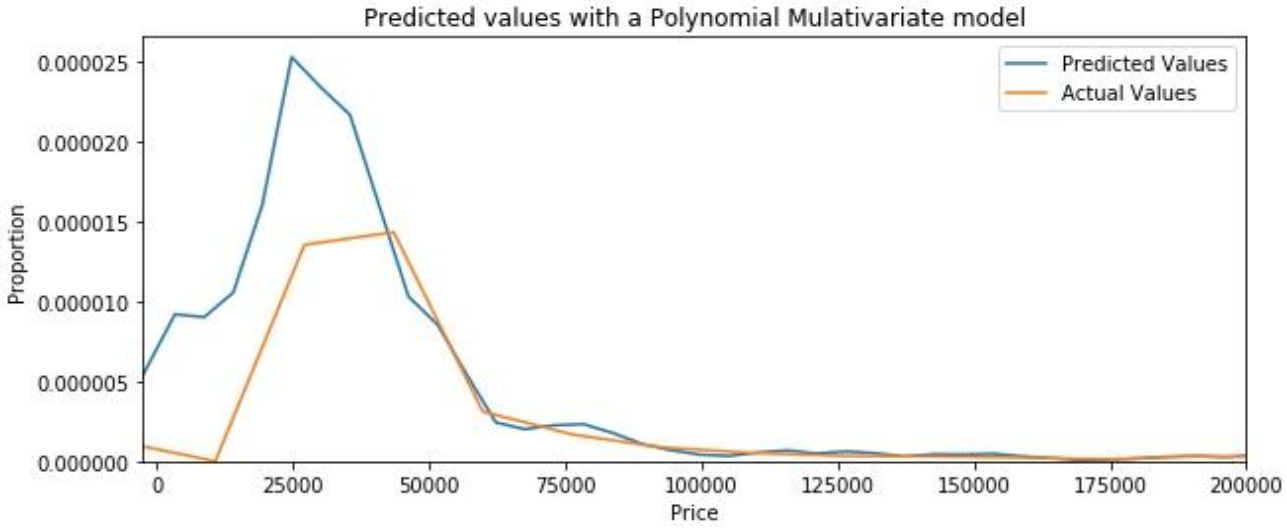

Рис. 4.25. Розподіл фактичних і прогнозованих значень

Проаналізувавши графік розподілу фактичних і прогнозованих значень, ми можемо зробити висновок, що в цілому найкращою моделлю для прогнозування цін на автомобілі є множинна поліноміальна регресія, оскільки вона є найточнішою із побудованих нами моделей.

Висновок до розділу 4

У даному розділі дипломного проекту нами були побудовані моделі для прогнозування цін на автомобілі. Для цього були використані такі алгоритми машинного навчання, як лінійна та поліноміальна регресія. У результаті аналізу точностей моделей та графіків розподілу фактичних і прогнозованих значень нами було встановлено, що найбільш точною моделлю прогнозування є множинна поліноміальна регресія, яка показала 84,79% точності на контрольній вибірці.

ВИСНОВКИ

У даному дипломному проекті була розроблена система моніторингу динаміки ринку. Дана система широко використовуються у сфері маркетингових досліджень ринку продажів.

У результаті виконання проекту було проведено дослідження ринку продажів автомобілів та, за допомогою інструментів для роботи з аналізом даних, були сформовані такі аналітичні дані як:

- залежність ціни від потужності двигуна;
- розподіл цін проданих автомобілів;
- залежність діапазону цін від розміру двигуна;
- матриця кореляції числових ознак, які характеризують автомобіль;

Також, за допомогою алгоритмів машинного навчання були побудовані моделі для прогнозування цін на автомобілі. Найбільш точною виявилась множинна поліноміальна модель, яка може спрогнозувати ціну автомобіля з точністю до 84,79%.

У даний час система моніторингу ринку є невід'ємною частиною управління бізнесом. Застосування методів машинного опрацювання інформації дозволяє виконувати аналіз даних та проводити прогнозування. На підставі одержуваних даних робляться висновки щодо основних тенденцій на ринку та прогноз перспектив.

У першому розділі дипломного проекту було розглянуто поняття маркетингового моніторингу ринку, його схему, структуру, цілі і задачі, основні етапи. Також було проаналізовано необхідність маркетингового моніторингу ринку в управлінні бізнесом.

У другому розділі було розглянуто машинне навчання та його основні поняття і позначення: модель алгоритмів, методи навчання, функціонали якості. Також проведено огляд основних алгоритмів машинного навчання та проаналізовано можливі галузі застосування методів машинного навчання.

У третьому розділи, нами було проведено огляд основних інструментів реалізації нашої системи серед яких: мова програмування Python, бібліотека для реалізації алгоритмів машинного навчання scikit-learn, бібліотека для обробки і аналізу даних pandas, платформу для наукових досліджень Anaconda.

У четвертому розділі ми перейшли до практичної реалізації системи, був проведений аналіз даних продажів автомобілів, встановлено залежності цін від ознак автомобіля та була побудована модель для прогнозування цін на транспортні засоби на базі алгоритму машинного навчання. Найбільш точним виявився алгоритм множинної поліноміальної регресії, який може прогнозувати ціну автомобіля з точністю до 84%.

Отже, результати робіт, проведених в рамках виконання дипломного проекту, покликані оцінити можливість застосування методів машинного навчання в сфері аналізу і прогнозу динаміки ринку продажу.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Федорович Р. В. Маркетинговий аналіз кон’юнктури ринку, 2009. – 110 с.
2. Н.В. Карпенко Маркетингова діяльність підприємств: сучасний зміст, Київ: Центр учбової літератури, 2016. – 252 с.
3. Черненко О.В. Маркетингова інформаційна система: механізм управління потоками, НТУУ «КПІ», 2012 – 138 с.
4. Луїс Педро Коельо, Віллі Річард, Построение систем машинного обучения на языке Python, 2016. – 302 с.
5. Guido S. Introduction to Machine Learning with Python / S. Guido, A. Müller. – Sebastopol, United States: O'Reilly Media, Inc, USA, 2016. – 392 с.
6. Lavreniuk M.S. Large-scale classification of land cover using retrospective satellite data, M.S. Lavreniuk, S.V. Skakun, A.J. Shelestov, 2016. – 138 с.
7. Kussul N. Parcel-based crop classification in ukraine using landsat-8 data and senti- nel-1A data, N. Kussul, G. Lemoine, F.J. Gallego, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2016. – 2500 – 2508 с.
8. Chen Y. Deep learning-based classification of hyperspectral data / Y. Chen, Z. Lin, X. Zhao, IEEE Journal of Selected topics in applied earth observations and remote sensing, 2014. — 2094 – 2107 с.
9. Zhao W. Learning multiscale and deep representations for classifying remotely sensed imagery, W. Zhao, S. Du, ISPRS Journal of Photogrammetry and Remote Sensing, 2016. — 155–165 с.
10. Kussul N.N. Land cover changes analysis based on deep machine learning technique, N.N. Kussul, N.S. Lavreniuk, A.Y. Shelestov, Journal of Automation and Information Sciences, 2016. — 42–54 с.
11. Kussul N. Geospatial Intelligence and Data Fusion Techniques for Sustainable De- velopment Problems, N. Kussul, A. Shelestov, R. Basarab, ICTERI. 2015.

— 196–203 с.

12. Markou M., Singh S. Novelty detection: A Review, Part 2: Neural Network-based Approaches, Signal Processing, 2003. — 2481–2497 с.
13. Markou M. Novelty detection: A Review, Part 1: Statistical Approaches, Signal Processing, 2003. — 2481–2497 с.
14. Peacock A., Ke X., Wilkerson M. Typing patterns: A key to user identification, IEEE Security and Privacy, 2004 — 40–47 с.
15. Ramaswamy S., Rastogi R., Shim K. E□cient Algorithms for Mining Outliers from Large Data Sets, ACM SIGMOD Conference, 2000. — 427–438 с.

ДОДАТОК А ЛІСТИНГ ПРОГРАМНОГО КОДУ

```
import pandas as pd
from tabulate import tabulate
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns

from scipy import stats
from sklearn.metrics import mean_squared_error
from sklearn.metrics import r2_score

from sklearn.model_selection import cross_val_predict
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import train_test_split

from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler

from matplotlib.ticker import FormatStrFormatter
import matplotlib.ticker as ticker

import warnings
warnings.filterwarnings('ignore')


### Підготовка даних ###


df = pd.read_csv(r'cars.csv')
df.rename(columns={'MSRP':'Price'}, inplace=True)

# Список даних
print(df.head())

# Найдорожчі автомобілі
print(df.nlargest(5, 'Price'))

# Список параметрів
print(df.info())

# Відповідні dtypes, які допомагають зменшити використання пам'яті
df = df.astype({'Engine Fuel Type':'category', 'Transmission Type':'category',
                'Vehicle Size':'category', 'Vehicle Style':'category'})

# Розподіл відсутніх даних
print('\n', df.isnull().sum())

# Список назв моделей з відсутнім ринковою категорією
missing_cat = df[df['Market Category'].isnull()]['Model'].drop_duplicates().tolist()


# Режим ринкової категорії кожної моделі
def cat_mode(model):
    mode = df[df['Model'] == model]['Market Category'].mode()
    if not (mode.empty):
        return mode.values[0]


# Модель та її найпоширенішу категорія
mode = {}
for model in missing_cat:
    mode.update({model: cat_mode(model)})
```

```
new_mode = {k: v for k, v in mode.items() if v is not None}

# Заміна кожного NaN на категорію режиму ринку
for model in new_mode:
    values = new_mode[model]
    df[df['Model'] == model] = df[df['Model'] == model].replace(np.nan, mode[model])

# Приклад найпоширенішої категорії для Honda Civic
print('\n', 'Most Honda Civic are', cat_mode('Civic'), 'vehicles', '\n')

# Відсутнє значення тип палива
print(df[df['Engine Fuel Type'].isnull()])

# Заміна значенням (звичайний неетильований)
df['Engine Fuel Type'].replace(np.nan, 'regular unleaded', inplace=True)

# Видалення електричних транспортних засобів
df.drop(df[(df['Engine Fuel Type']=='electric')].index, inplace=True)

# Значення кількісті циліндрів роторних двигунів замінюємо на середньостатистичне
print(df[df['Engine Cylinders'].isnull()].head())
engine_mean = round(df['Engine Cylinders'].astype('float').mean())
df['Engine Cylinders'].replace(np.nan, engine_mean , inplace=True)

# Значення кількості дверей для Ferrari FF замінюємо на 2
print(df[df['Number of Doors'].isnull()])

# По аннології до категорії ринку замінюємо значення потужності двигуна на середнє
print(df[df['Engine HP'].isnull()].head())

# Список назв моделей з відсутнім двигуном hp
missing_hp = df[df['Engine HP'].isnull()]['Model'].drop_duplicates().tolist()


# Середній  hp кожної моделі
def engine_mean(model):
    mean = round(df[df['Model'] == model]['Engine HP'].mean())
    return mean


# Замінити значення Nan на середнє значення
for model in missing_hp:
    df[df['Model'] == model] = df[df['Model'] == model].replace(np.nan, engine_mean(model))

# Приклад для Lincoln Continental
print('\n', 'The average Lincoln Continental power is', '{0:.0f}'.format(engine_mean('Continental')), 'HP', '\n')

# Список hp двигуна для Impala
hp_impala = df['Engine HP'].loc[df['Model']=='Impala'].tolist()

# Відокремимо числові та категоріальні значення
num_hp = [value for value in hp_impala if isinstance(value, float)]
cat_hp = [cat for cat in hp_impala if isinstance(cat, str)]

# Середня потужність
mean_hp_impala = round(np.mean(num_hp))

# Замінимо помилкові значення на середні
df.loc[df['Model']=='Impala'] = df[df['Model']== 'Impala'].replace(cat_hp[0], mean_hp_impala)

# Набір даних без відсутніх значень
print(df.isnull().sum(), '\n')

# Додаємо параметр віку авто
df['Age'] = 2017 - df['Year']
```

```
# Переводимо споживання палива у літр на 100 км
df[['highway MPG', 'city mpg']] = 235 / df[['highway MPG', 'city mpg']]

# Перейменовуєм нові стовпці та округлюєм їх
df.rename(columns={'highway MPG':'Highway L/100km', 'city mpg':'City L/100km'}, inplace=True)
df[['Highway L/100km', 'City L/100km']] = df[['Highway L/100km', 'City L/100km']].round(decimals=2)

# Підготовлений дата сет
print(df.head())


### Дослідження даних ###


# Залежність ціни від потужності двигуна
fig, ax = plt.subplots(figsize=(10, 5))

ax.scatter(df['Engine HP'], df['Price'], s=1, alpha=0.7)

# Установим назви для осей
plt.title('Price with respect to Engine HP')
plt.xlabel('Engine Horsepower')
plt.ylabel('Price')

# Роздільник для тисяч осей у
ax.yaxis.set_major_formatter(ticker.StrMethodFormatter('{x:,.0f}'))

plt.show()

# Розбиваємо дані для візуалізації даних
df_viz = df.loc[df['Price'] <= 100000]
df_hypercar = df.loc[df['Price'] > 100000]

# Розподіл цін
plt.figure(figsize=(10, 4))
plt.hist(df_viz['Price'], bins=80)

plt.title('Price distriution (for vehicles < 100K $)')
plt.xlabel('Price in $')
plt.ylabel('count')
plt.show()

# Діапазон ціни в залежності від розміру двигуна
fig, axes = plt.subplots(1, 2, figsize=(12,4))

# Робимо 2 ділянки із ціновою категорією
sns.boxplot(x='Engine Cylinders', y='Price', data=df_viz,
        ax=axes[0], palette="spring").set_title('Vehicles < 100K$')

sns.boxplot(x='Engine Cylinders', y='Price', data=df_hypercar,
        ax=axes[1], palette="PuRd").set_title('Vehicles > 100K$')

plt.tight_layout()
plt.subplots_adjust(wspace = 0.4)
plt.show()

# 2 найдешевші та найдорожчі 12-циліндрові машини
print(df_viz[df_viz['Engine Cylinders'] == 12].iloc[np.r_[0:2, -2:0]])

# Співвідношення кореляції між змінними
corr_map = df.corr()

# Маска для верхнього трикутника
mask = np.zeros_like(corr_map, dtype=np.bool)
mask[np.triu_indices_from(mask)] = True

# Намалюємо теплову карту
```

```
f, ax = plt.subplots(figsize=(10, 5))

sns.heatmap(corr_map, mask=mask, cmap='RdBu_r', vmax=1, center=0,
            square=True, linewidths=.5, ax=ax)

plt.xticks(rotation='60')
plt.title('Heatmap correlation of the numerical variables')

plt.show()

# Таблиця кореляції
print(df.corr())

df = df.drop(columns=['Year', 'Highway L/100km'], axis=1)

# Спостереження взаємозв'язків між змінними
sns.pairplot(df)

plt.show()

# Логарифм трансформації ціни
df['Log_price'] = df['Price'].apply(lambda x : np.log(x+1))

# Потужність
df['Sq_engine_hp'] = df['Engine HP'].apply(lambda x : np.square(x))
df['Sqrt_engine_hp'] = df['Engine HP'].apply(lambda x : np.sqrt(x))
df['Log_engine_hp'] = df['Engine HP'].apply(lambda x : np.log(x+1))

# Таблиця кореляції
corr = df.corr()
print(corr[['Price', 'Log_price']].loc[['Engine HP', 'Sq_engine_hp', 'Sqrt_engine_hp', 'Log_engine_hp'], :])

# Вік
df['Sq_age'] = df['Age'].apply(lambda x : np.square(x))
df['Sqrt_age'] = df['Age'].apply(lambda x : np.sqrt(x))
df['Log_age'] = df['Age'].apply(lambda x : np.log(x+1))

# Таблиця кореляції
corr = df.corr()
print(corr[['Price', 'Log_price']].loc[['Age', 'Sq_age', 'Sqrt_age', 'Log_age'], :])

# Розхід пального у місті L/100km
df['Sq_city'] = df['City L/100km'].apply(lambda x : np.square(x))
df['Sqrt_city'] = df['City L/100km'].apply(lambda x : np.sqrt(x))
df['Log_city'] = df['City L/100km'].apply(lambda x : np.log(x+1))

corr = df.corr()
print(corr[['Price', 'Log_price']].loc[['City L/100km', 'Sq_city', 'Sqrt_city', 'Log_city'], :])

# Відкинемо невикористані стовпці
df = df.drop(['Sq_engine_hp', 'Sqrt_engine_hp', 'Sqrt_age', 'Log_age', 'Sqrt_city', 'Log_city'], axis=1)

print('\n', df.loc[:,['Log_price', 'Log_engine_hp', 'Sq_age', 'Sq_city']].head(), '\n')


### Моделювання ###


# Проста лінійна регресія

# Визначення X і цільової змінної Y
X = df[['Log_engine_hp']]
Y = df[['Log_price']]

# Розділення набору даних в тренувальний і тестовий набір
# Ми будемо використовувати 80% набору даних для навчання і 20% для тестування
xtrain, xtest, ytrain, ytest = train_test_split(X, Y, test_size=0.2, random_state=53)
```

```
# Проста лінійна регресія з одним вхідним параметром Log_engine_hp
lr = LinearRegression()

lr.fit(xtrain, ytrain)

print('The slope is ', '{0:.4f}'.format(lr.coef_.item()))

print('The intercept is ', '{0:.4f}'.format(lr.intercept_.item()), '\n')

# Графік лінії прогнозування та її залишкової помилкі
fig, (ax1, ax2) = plt.subplots(ncols=2, sharey=False, figsize=(12, 5))

ax1= sns.regplot(x='Log_engine_hp', y='Log_price', data=df, scatter_kws={"s": 5}, ax=ax1)
ax1.set_title('Regression line of the Engine Cylinders')

ax2 = sns.residplot(X, Y, scatter_kws={"s": 3}, ax=ax2)
ax2.set_title('Residual plot')
ax2.set_ylabel('')
ax2.set_ylim(-5, 5)

plt.tight_layout()
plt.show()

# Зробимо перехресну перевірку на 4 підмножини
score = cross_val_score(lr, xtrain, ytrain, cv=4)

print('The cross-validations sets R-squared are :', score)
print('The average performance is therefore', np.mean(score), '\n')

# Зробимо деякі прогнози
yhat_simple = cross_val_predict(lr, xtest, ytest, cv=4)
print('The 1st prediction is', round(np.exp(yhat_simple[0].item())),
      'whereas the actual value was', df.Price[0], '\n')

# Оцінка моделі
simple_linear_R2_train = lr.score(xtrain, ytrain)
simple_linear_R2_test = lr.score(xtest, ytest)

# Ми трансформуємо yhat_simple для виконання тесту MSE
s_linear_MSE = mean_squared_error(ytest, np.exp(yhat_simple))

print('Simple Linear Regression', '\n')

print('Mean Squared Error', s_linear_MSE)
print('R-squared on train data', simple_linear_R2_train)
print('R-squared on test data', simple_linear_R2_test, '\n')

# Графік розподілу фактичних і прогнозованих значень
plt.figure(figsize=(10, 4))

ax1 = sns.distplot(yhat_simple, hist=False, label='Predicted Values')
sns.distplot(df['Log_price'], hist=False, label='Actual Values', ax=ax1)

plt.ylabel('Probability')
plt.legend()
plt.title('Predicted values using Simple Linear Regression')

plt.show()

# Поліноміальна регресія

# Функція для графічного зображення апроксимації поліноміальної регресійної моделі
def PlotPoly(model, independant_var, dependant_var):
    # Calculate the polynomial function for 100 points
    new_x = np.linspace(min(df[independant_var]), max(df[independant_var]), 1000)
    y_model = model(new_x)
```

```
    # Graph of the data points and the polynomial line
    plt.figure(figsize=(10, 5))
    plt.plot(X[independant_var], df[dependant_var], '.', markersize=1)
    plt.plot(new_x, y_model, '-')

    # Axis label and x-axis setting
    plt.xlabel(independant_var)
    plt.ylabel('Price')

    plt.show()
    plt.close()

# Встановлюємо змінні
X = df[['Engine HP']]
Y = df[['Price']]
xtrain, xtest, ytrain, ytest = train_test_split(X, Y, test_size=0.3, random_state=53)

# Обчислюємо поліноміальну функцію порядку 3
f = np.polyfit(xtrain['Engine HP'], ytrain['Price'], 3)
p = np.poly1d(f)  # наша модель

print('Polynomial function of the Engine Horsepower :', '\n')
print(p, '\n')

# Прогнозування
yhat_simple_poly = p(xtest)

print('Prediction', round(yhat_simple_poly[0].item()))
print('Actual value', df['Price'][0], '\n')

# Оцінка моделі
poly_R2_test = r2_score(ytest, yhat_simple_poly)

s_poly_MSE = mean_squared_error(ytest, yhat_simple_poly)

print ('Polynomial Regression')
print('R-square on the test data', poly_R2_test)
print('Mean Squared Error', s_poly_MSE, '\n')

# Графік лінії поліноміальної регресії
PlotPoly(p, 'Engine HP', 'Price')

# Графік розподілу фактичних і прогнозованих значень
plt.figure(figsize=(10, 4))

ax1 = sns.distplot(yhat_simple_poly, hist=False, label='Predicted Values')
sns.distplot(Y, hist=False, label='Actual Values', ax=ax1)

# Встановлюємо назви осей та мітки
plt.title('Predicted values using Polynomial regression')
plt.xlim(-2500, 300000)
plt.xlabel('Price')
plt.ylabel('Probability')
plt.legend()

plt.show()

# Множинна лінійна регресія

# Лінійна регресія з кількома параметрами
lm = LinearRegression()

Z = df[['Log_engine_hp', 'Sq_age', 'Sq_city']]
Y = df['Log_price']

# Розподіляємо дані для навчання та тестування
```

```
ztrain, ztest, ytrain, ytest = train_test_split(Z, Y, test_size=0.3, random_state=53)

lm.fit(ztrain, ytrain)
print('Multiple Linear Regression', '\n')
print('Coefficient for each variable', lm.coef_)
print('Intercept term :', lm.intercept_, '\n')

# Прогнозування
yhat_multi = lm.predict(ztest)
print('Prediction :', round(np.exp(yhat_multi[0].item()), 0))
print('Actual value:', df.Price[0], '\n')

# Оцінка моделі
m_linear_R2 = lm.score(ztrain, ytrain)
m_linear_MSE = mean_squared_error(ytest, np.exp(yhat_multi))

print('Multiple Linear Regression')
print('Mean Squared Error', m_linear_MSE)
print('R-squared', m_linear_R2, '\n')

# Графік розподілу фактичних і прогнозованих значень
plt.figure(figsize=(10, 4))

ax1 = sns.distplot(yhat_multi, hist=False, label='Predicted Values')
sns.distplot(Y, hist=False, label='Actual Values', ax=ax1)

plt.ylabel('Proportion')
plt.legend()
plt.title('Predicted values using Multiple variables')

plt.show()

# Множинна поліноміальна регресія

# Встановлюємо параметри
Z = df[['Engine HP', 'Age', 'City L/100km']]
Y = df['Price']

ztrain, ztest, ytrain, ytest = train_test_split(Z, Y, test_size=0.3, random_state=53)

# Створення та навчання моделі
Input = [('scale', StandardScaler()), ('polynomial', PolynomialFeatures(degree=3)), ('model', LinearRegression())]

pipe = Pipeline(Input)
pipe.fit(ztrain, ytrain)

# Прогнозування
yhat_poly_multi = pipe.predict(ztest)

print('Prediction :', round(yhat_poly_multi[0].item(), 0))
print('Actual value :', df.Price[0], '\n')

# Оцінка моделі
m_poly_R2 = r2_score(ytest, yhat_poly_multi)
m_poly_MSE = mean_squared_error(ytest, yhat_poly_multi)

print('Multivariable Polynomial Regression')
print('Mean Squared Error', m_poly_MSE)
print('R-square', m_poly_R2, '\n')

# Графік розподілу фактичних і прогнозованих значень
plt.figure(figsize=(10, 4))

ax1 = sns.distplot(yhat_poly_multi, hist=False, label='Predicted Values')
sns.distplot(df['Price'], hist=False, label='Actual Values', ax=ax1)

plt.xlim(-2500, 200000)
```

```
plt.ylabel('Proportion')
plt.legend()
plt.title('Predicted values with a Polynomial Mulativariate model')

plt.show()
```

ДОДАТОК Б КАЛЕНДАРНИЙ ПЛАН